

# Estimation et réduction du coût d'algorithmes quantiques

par

Jessica Lemieux

Thèse présentée au département de physique  
en vue de l'obtention du grade de docteur ès science (Ph.D.)

FACULTÉ des SCIENCES  
UNIVERSITÉ de SHERBROOKE

Sherbrooke, Québec, Canada, 3 avril 2022

Le 31 mars 2022

*le jury a accepté la thèse de Mme Jessica Lemieux dans sa version finale.*

Membres du jury

Professeur David Sénéchal  
Directeur de recherche  
Département de physique

Professeur Guillaume Duclos-Cianci  
Directeur de recherche  
Département de physique

Professeur Rolando Somma  
Membre externe  
Department of Physics and Astronomy, University of New Mexico  
*et* Los Alamos National Laboratory

Professeur Jeffrey A. Quilliam  
Membre interne  
Département de physique

Professeur Stefanos Kourtis  
Président rapporteur  
Département de physique

À David Poulin

# Sommaire

Plusieurs algorithmes ont été marquants dans le développement de l'informatique quantique : on peut penser notamment à l'algorithme de Shor ou à l'algorithme de Grover. Bien que pour ce dernier, par exemple, nous ayons une accélération quadratique prouvée, il n'est pas clair que ce sera suffisant pour offrir un avantage pratique par rapport aux algorithmes classiques. D'abord, celui-ci est formulé en termes d'oracle, une boîte noire qui cache une sous-routine non comprise dans le calcul du coût. Ensuite, lorsque l'on prend en considération le surcoût engendré notamment par la correction d'erreur quantique, il est possible de perdre l'accélération promise. Mais également, la durée d'une porte logique quantique est considérablement plus long que son homologue classique. Quel est le coût réel d'un algorithme quantique lorsque l'on prend en considération toutes les sous-routines ? À quand un ordinateur quantique utile qui surpassera les performances d'un superordinateur ?

Dans cette thèse, nous présenterons trois projets visant tous à estimer et réduire le coût d'algorithmes ou sous-routines quantiques. Les algorithmes abordés sont issus d'une discrétisation de l'évolution adiabatique. Le premier se concentre sur un algorithme de préparation d'état d'un système à  $N$  corps par une évolution adiabatique via l'effet Zénon. Le second porte sur une version quantique des algorithmes de marches aléatoires et de recuit simulé pouvant, par exemple, préparer un état stationnaire. Le dernier décrit un nouvel algorithme : une évolution adiabatique basée sur la réflexion. Celui-ci permet, entre autres, de résoudre des problèmes MAX- $k$ SAT, une classe de problèmes NP-difficile. Avec ces projets, nous voulons, d'une part, proposer des algorithmes efficaces ainsi que leur implémentation de A à Z et, d'autre part, estimer les caractéristiques nécessaires à un ordinateur quantique utile (p. ex. taille, résistance au bruit, vitesse d'opération).

Les résultats présentés démontrent le coût élevé associé aux algorithmes tolérants aux fautes. Bien qu'on s'attende à avoir une accélération par rapport au classique, lorsque l'on prend en considération le nombre de qubits physiques, le nombre d'opérations physiques et la durée de chacune de ces opérations, en incluant la correction d'erreur notamment, la taille des instances offrant un avantage réel est loin d'être atteignable pour les processeurs quantiques à court terme. Toutefois, en combinant

des méthodes astucieuses et au moyen de différents procédés d'optimisation, il est possible de réduire considérablement le coût des algorithmes quantiques, et donc de réduire le délai pour atteindre la suprématie quantique.

# Remerciements

J'ai eu la chance de me retrouver au bon endroit, au bon moment. L'Université de Sherbrooke et l'Institut Quantique sont des milieux stimulants qui m'ont apporté une multitude d'occasions pour grandir en tant que scientifique. J'ai eu la chance de participer à des stages, des conférences et des écoles de pointe en informatique quantique, mais surtout, j'ai eu la chance d'être incroyablement bien entourée. Il y a l'entourage académique bien sûr, tous ces chercheurs de renom que nous avons ici à Sherbrooke, ou ces scientifiques que j'ai pu rencontrer en école d'été et en congrès, mais il y a également toutes ces personnes qui ont été d'un soutien technique et moral. Voici quelques une des personnes qui forger mon parcours.

Tout d'abord, je voudrais remercier mon superviseur, David Poulin, à qui je dédit cette thèse. Le 25 juin 2020 marque une grande perte pour la communauté en informatique quantique. David était un scientifique généreux de son temps et de ses idées, jusqu'à la fin. Il avait à coeur la réussite de ses étudiants. L'avoir comme mentor, c'est avoir la chance d'accélérer son train d'apprentissage, d'exploiter son plein potentiel, avec l'espace pour développer ses propres ambitions et sa personnalité de chercheuse, le tout, avec humour. Si j'arrive à devenir ne serait-ce qu'une fraction du scientifique et de l'homme qu'il était, ma vie aura été un succès. Je crois parler au nom de tout ceux qui eux la chance de te cotoyer David en te disant *merci pour tout*.

Je voudrais ensuite remercier les professeurs David Sénéchal et Guillaume Duclos-Cianci qui on bien voulu prendre la suite de ma supervision. David, merci pour ta disponibilité et ta rigueur. Guillaume, merci pour ta patience, ton professionnalisme et ton soucis du détail. Merci également à Stefanos Kourtis, mon président rapporteur, de m'avoir accueilli et inclus dans son groupe.

À tous les gens de mon groupe de recherche, qui ont partagé mon espace de travail et mon quotidien, avec qui j'ai eu des discussions enrichissantes et éclairantes merci. Merci à Andrew Darmawan, Anirban Narayan Chowdhury, Anirudh Krishna, Benjamin Bourassa, Christopher Chubb, Colin Trout, Guillaume Dauphinais, Maxime Tremblay, Nouédyn Baspin, Pavithran Iyer, Thomas Gobeil et Ye-hua Liu. Je voudrais remercier plus particulièrement Benjamin, un ami et partenaire d'étude, ainsi que

Maxime, qui est toujours d'une grande aide, autant sur le plan scientifique et technique que sur le plan moral.

Merci à mes parents Joanne et Robert pour leur soutien constant. Merci à Jean-Lou, mon partenaire de vie, qui m'encourage depuis l'adolescence à persévérer. Je vous dit merci aussi, Joanne et Jean-Lou, pour toutes les relectures au cours des dernières années (dont cette thèse!) Je vous aime.

Merci à mes amis pour leur support, notamment Alexandre Choquette, Andréanne Bombardier, Boris Cvjetkovic, Claude Rohrbacher, Jean-Michel Naud, Laurine Marian, Louis Häberle, Marie-Eve Boulanger, Martin Schnee, Oumar Kaba, Pierre Olivier Downey, Samuel Desrosiers et Sara Robinson.

Les mots me manque pour exprimer l'ampleur de ma gratitude. À vous tous, je dis humblement merci.

# Table des matières

<b>Sommaire</b>	<b>iii</b>
<b>Introduction</b>	<b>1</b>
<b>1 Motivation et fondement du projet</b>	<b>6</b>
1.1 Coût de l'ordinateur quantique . . . . .	6
1.2 Accélération quantique . . . . .	11
1.3 Évolution adiabatique . . . . .	13
1.4 Algorithme de Grover . . . . .	15
1.5 Algorithme adiabatique de Grover . . . . .	18
1.6 Effet Zénon quantique . . . . .	20
1.7 Récapitulatif et présentation des projets . . . . .	23
<b>2 Préparation d'état par évolution adiabatique discrète</b>	<b>25</b>
2.1 Article . . . . .	27
<b>3 Algorithme de marche aléatoire</b>	<b>36</b>
3.1 Article . . . . .	37
<b>4 Algorithme adiabatique basé sur la réflexion</b>	<b>53</b>
4.1 Article . . . . .	54
<b>Conclusion</b>	<b>65</b>
<b>A Matériel supplémentaire</b>	<b>68</b>
A.1 Algorithme d'estimation de phase . . . . .	68
A.2 Le modèle de Hubbard . . . . .	71
A.3 Marche aléatoire classique . . . . .	73
A.4 Opérateur de marche de Szegedy . . . . .	76
<b>Bibliographie</b>	<b>78</b>



# Liste des figures

1	Loi de Moore	2
1.1	Niveaux de l'ordinateur quantique	7
1.2	Circuit d'injection	9
1.3	Itération de l'algorithme de Grover	16
1.4	Circuit de l'algorithme de Grover	18
A.1	Algorithme d'estimation de phase : le retour de phase	69
A.2	Transformée de Fourier quantique	70
A.3	Estimation de phase quantique	71
A.4	Chaîne de Markov	75

# Liste des tableaux

1.1 Définition des accélérations quantiques . . . . .	12
---	----

# Introduction

*Let us stop trying to explain what this means, but try to do something with it. It is weird. It obeys different laws of probabilities, different laws of logic, and computation is all about probabilities and logic. So if we build computers out of these systems, maybe, you know, the rules would change and indeed they do.*

– Poulin, 2018 [1]

L'ordinateur classique programmable fait son apparition dès le début de la décennie de 1940 [2]. Cet outil fait boule de neige. En effet, que ce soit pour la modélisation de systèmes complexes, pour tester des théories ou pour la synthèse d'images, il est clair que cette technologie facilite les avancements dans une multitude de domaines. C'est l'avènement du transistor, en 1947, qui déclenche une progression monstre en informatique. Présente dans tous les dispositifs électroniques modernes, cette composante fondamentale des circuits électriques permet entre autres de construire les portes logiques ou encore d'amplifier le signal électrique. Depuis, le transistor devient de plus en plus performant, petit et abordable. Le boum de développement engendré par celui-ci nous a permis d'obtenir les ordinateurs d'aujourd'hui toujours plus petits et plus puissants. Connue sous le nom de loi de Moore [3], et observée pour la première fois par Gordon E. Moore en 1965, on remarque qu'à chaque deux ans tout au plus, les puces électroniques contiennent deux fois plus de transistors, toujours plus efficaces que les précédents. Toutefois, cette tendance a ses limites : nous avons atteint la taille à laquelle les effets de la mécanique quantique sont non négligeables. Si nous voulons accroître notre potentiel de calculs, il faut maintenant augmenter le nombre de composantes en augmentant nécessairement aussi la taille de l'ordinateur — à moins, bien sûr, de concevoir un nouveau type d'appareil nous

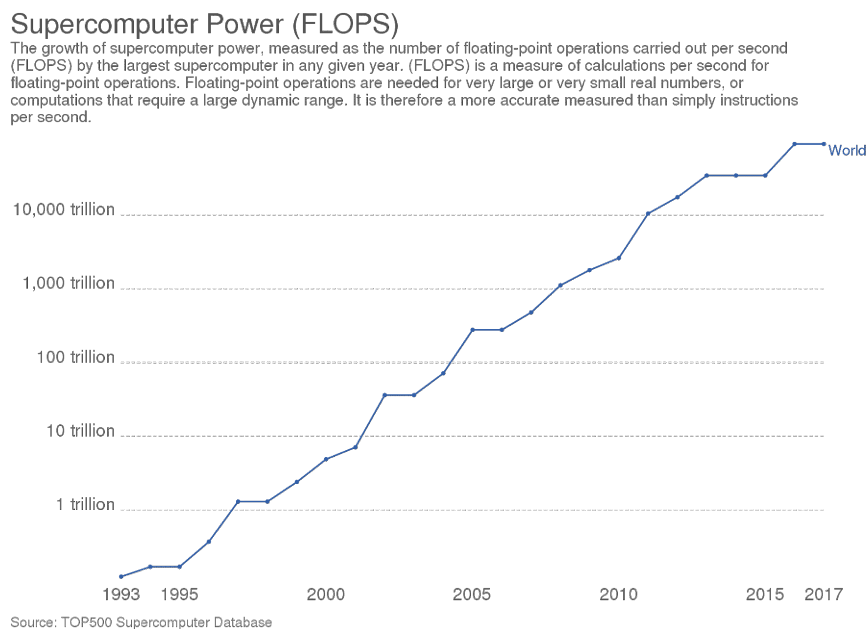


FIGURE 1 – Croissance de la puissance des supercalculateurs, mesurée par le nombre d’opérations à virgule flottante effectuées par seconde (FLOPS) [4]

permettant éventuellement d’avoir un outil dont les composantes seraient en deçà de la limite classique-quantique actuelle.

Toutefois, le progrès matériel n’est pas l’unique responsable de l’augmentation des performances dans le domaine informatique. Au contraire, les coûts algorithmiques ont des conséquences majeures sur l’efficacité. Il est évident que si, pour une même tâche, deux algorithmes ont des différences notables de complexité, l’algorithme de complexité inférieure aura un avantage sur le second. À tout le moins, pour les problèmes de grandes tailles. De manière plus surprenante, si l’on utilisait un algorithme moderne sur un système d’autrefois, les performances seraient supérieures à celles d’un algorithme désuet exécuté sur un système à la fine pointe de la technologie. Le nombre d’opérations effectuées par seconde n’est qu’un facteur sur le temps total d’un calcul. En 1993, l’appareil le plus puissant pouvait effectuer 124 milliards d’opérations à virgule flottante par seconde (FLOPS) alors qu’en 2020, on parle de 442 mille milliards de FLOPS, soit un dispositif près de 3.5 millions de fois plus performant (voir la figure 1). Bien que le ratio soit impressionnant, il n’en reste pas moins qu’un algorithme nécessitant  $2^n$  opérations sur le plus récent superordinateur serait rapidement surpassé par un algorithme quadratiquement plus rapide (nécessitant  $2^{n/2}$

opérations), même si ce dernier était effectué sur un supercalculateur de 1993. En fait, dans cet exemple, dès que  $n \geq 44$ , la seconde option se révèle la plus efficace.

La mécanique quantique a été introduite au début du 20e siècle pour tenter d'expliquer les phénomènes connus aujourd'hui comme le rayonnement du corps noir, l'effet photoélectrique et l'effet Compton [5]. Ces expériences furent les premières à esquisser la nécessité de la discrétisation de certaines quantités physiques comme l'énergie et le moment cinétique. Cette nouvelle vision de la physique de l'infiniment petit devait ouvrir les portes à de nouveaux domaines de recherche, notamment l'informatique quantique.

Les technologies quantiques promettent de résoudre des problèmes hors d'atteinte pour nos appareils actuels. Bien qu'on puisse imaginer la continuation de la progression des processeurs et l'augmentation de la mémoire, il existe tout de même une limite infranchissable du point de vue pratique. L'information quantique exige toutefois un renouveau dans nos méthodes de résolution de problèmes. Les algorithmes ne seront plus alors basés simplement sur la logique binaire conventionnelle comme leurs homologues classiques.

Au début des années 80, Feynman fut le premier à proposer une nouvelle forme d'ordinateur, un ordinateur basé sur les principes de la mécanique quantique [6]. En mettant à profit ses particularités non déterministes, telles que la superposition d'états et l'intrication, Feynman affirmait déjà à l'époque qu'un tel outil pourrait simuler de manière efficace un système quantique ;

*Can a quantum system be probabilistically simulated by a classical [ ... ] universal computer? [...] No! [Y]ou must directly generate the probabilities [,] we have no way to store all the numbers, we have to just imitate the phenomenon directly*  
 – Feynman, 1982 [6]

Quoi de mieux qu'un système quantique pour simuler un système quantique? Il obéit aux mêmes règles et évite la représentation vectorielle de sa fonction d'onde qui nécessite un coût exponentiel, en matière de stockage notamment. On estime qu'un ordinateur classique ne peut guère simuler la dynamique de systèmes quantiques au-delà de quelques dizaines de sites [7]. Le calculateur quantique permettra donc de

résoudre des problèmes étant autrement impossibles à solutionner par un appareil classique. Il va sans dire que ces simulations ne sont pas l'unique motivation.

Plusieurs algorithmes ont marqué la recherche dans les dernières décennies. On pense notamment à l'algorithme de Grover [8] qui possède une accélération quadratique démontrable. Ainsi qu'à celui de Shor [9] (algorithme de factorisation) ayant — pour l'instant — une accélération exponentielle si on le compare au meilleur algorithme classique connu. Ou encore aux algorithmes d'optimisation (NISQ [10]<sup>1</sup> ou non [12]) qui ont le potentiel d'offrir une meilleure solution que leur contrepartie classique. Sans oublier la simulation de systèmes quantiques où nous avons une accélération exponentielle espérée (ou *accélération potentielle*, voir Tab. 1.1). Par contre, chacun de ces exemples est formulé en termes d'oracles ou omet le coût de certaines sous-routines pouvant être cruciales. L'importance de la complexité algorithmique est indéniable, mais il ne faut tout de même pas négliger les préfacteurs, autant au niveau matériel que logiciel. Ceci est particulièrement important lorsqu'on considère les premières applications quantiques potentielles. Les premiers ordinateurs tolérant aux fautes seront probablement aussi limités en qubits et en nombre d'opérations. Dans ce contexte, un facteur deux peut être déterminant ; un problème d'une certaine taille pourra ou non être résolu avec la technologie accessible. L'optimisation, dans le détail, est donc nécessaire. Ainsi, la table est mise pour la question de recherche centrale de cette thèse :

Quel est le coût réel d'un algorithme quantique ?

Cette question permet d'établir les bases de réponses à une autre question de grand intérêt : à quand un ordinateur quantique utile ? Évidemment, on ne peut répondre à l'une ou l'autre de ces questions de manière générale. Le but premier de cette thèse est donc de se pencher sur des problèmes d'intérêt et d'étudier, dans le détail, le coût nécessaire à l'obtention d'une solution. Nous allons ainsi nous consacrer à trois algorithmes : un algorithme de préparation d'état, un algorithme de marche aléatoire quantique, et un algorithme d'optimisation. Les trois routines étudiées sont dérivées d'algorithmes ayant une accélération quadratique démontrable. Celles-ci, comme

---

1. L'appellation *NISQ* (de l'anglais, Noisy Intermediate-Scale Quantum [11]) est attribuée aux algorithmes qui pourront être réalisés à court ou moyen terme sur des ordinateurs quantiques de taille intermédiaire. Ces derniers, ayant des ressources plus limitées, n'utiliseront pas de code de correction d'erreur et, conséquemment, seront plus susceptibles au bruit, d'où le qualificatif *Quantique Bruyant à Échelle Intermédiaire*.

c'est le cas pour la plupart des accélérations quadratiques, peuvent être interprétées comme des cas particuliers de l'algorithme de Grover.

## Chapitre 1

# Motivation et fondement du projet

*Instead of looking at quantum systems purely as phenomena to be explained as they are found in nature, [pioneers of quantum information] looked at them as systems that can be **designed***

— Nielsen and Chuang, 2010 [13]

## 1.1 Coût de l'ordinateur quantique

Pour concevoir un ordinateur quantique, il faut tenir compte des différents niveaux, depuis les qubits physiques jusqu'à l'interface utilisateur (voir Fig. 1.1). Chaque niveau est construit en considérant les niveaux qui le précèdent. Pour comprendre la surenchère de ressources nécessaires à un ordinateur tolérant aux fautes, considérons les ressources requises du premier niveau jusqu'au dernier.

Le **premier niveau** est formé des bits quantiques permettant de stocker l'information. Souvent représenté sur une sphère de Bloch, l'état d'un qubit  $|\Psi\rangle$  est donné par une superposition dans une base orthonormale  $|\Psi\rangle = \sum_j \alpha_j |j\rangle$  où  $\alpha_j \in \mathbf{C}$  et  $\sum_j |\alpha_j|^2 = 1$ ;

$$\langle i | j \rangle = \begin{cases} 1 & \text{si } i = j \\ 0 & \text{sinon.} \end{cases} \quad (1.1)$$

La base communément utilisée, aussi appelée base de calcul, est la base  $|0\rangle$  et  $|1\rangle$ . Pour une introduction au domaine, consulter [13].



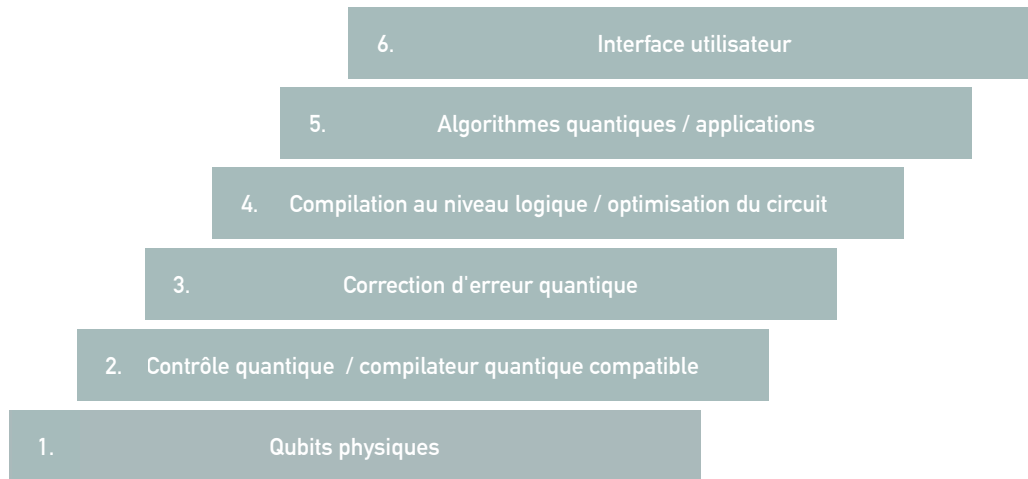


FIGURE 1.1 – Niveaux relatifs à la conception d'un ordinateur quantique

Le **second niveau** concerne le contrôle quantique permettant la manipulation de l'information. Cela nécessite notamment un compilateur qui traduit les commandes, ou portes quantiques, en opérations physiques (une impulsion sur un qubit de spin par exemple). À ce niveau, les portes quantiques sont données comme une séquence d'éléments de l'*ensemble universel* de portes. Cet ensemble discret permet de reproduire une opération unitaire quelconque sur les qubits à une précision  $\varepsilon$  arbitrairement petite. L'ensemble universel le plus couramment utilisé est Clifford+ $T$ . Il est constitué des générateurs<sup>1</sup> du groupe de Clifford, soient la porte de Hadamard ( $H$ ), racine de pauli  $Z$  ( $S$ ) et la porte à deux qubits Contrôle-Non (CNOT), représentées respectivement par les matrices

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}, \quad CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

1. Notons que l'ensemble des générateurs d'un groupe n'est pas unique. Pour cette thèse, nous donnons en exemple les portes  $H$ ,  $S$  et CNOT puisqu'elles sont plus couramment utilisées dans la littérature. Par contre, il convient d'utiliser un ensemble qui soit natif au système physique ; une porte aisément effectuée sur le système. Il est possible notamment de remplacer la porte à deux qubits CNOT par la porte C-Z.

La porte

$$T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$$

est la racine de la porte  $S$ . Ainsi, le compilateur physique traduit la séquence de portes par une série d'impulsions appropriées sur les qubits d'intérêts.

Le **troisième niveau** se consacre à la correction d'erreur quantique. Comme nous pouvons manipuler l'information, l'environnement le peut aussi. Il faut donc la protéger. Pour un ordinateur quantique tolérant aux fautes, on utilise un principe de redondance pour mieux préserver notre information. Grossièrement, l'idée est d'utiliser plusieurs qubits physiques, qui sont bruyants, pour encoder un qubit logique, qui aura ainsi une meilleure résistance aux fautes. En ajoutant des qubits physiques, on augmente la taille de l'espace de Hilbert accessible à notre système. Par contre, on restreint les états utilisés à un sous-espace défini par symétrie (ou propriété des états). Les opérateurs pourraient, par exemple, conserver la parité de l'état des qubits physiques. Ainsi, une mesure permet de vérifier si aucune erreur n'est introduite pendant une opération; elle certifie que l'état obtenu est toujours dans le sous-espace. Si ce n'est pas le cas, il faut idéalement déterminer et corriger l'erreur, le tout sans provoquer l'effondrement de la fonction d'onde logique. Les portes logiques devront donc avoir un équivalent en termes de portes physiques. Toutefois, une telle implémentation n'est pas simple : le théorème de Eastin et Knill stipule qu'il n'existe aucun code *stabilisateur*<sup>2</sup> qui possède un ensemble universel de *portes transversales*<sup>3</sup>, soit la porte  $T$  pour une majorité de codes. Il faut donc introduire d'autres méthodes permettant de conserver la tolérance aux fautes. Pour ce faire, on utilise l'injection et la distillation d'états magiques. Un état magique est un état qui, par injection, permet d'appliquer une porte sur le reste du système. Par exemple, l'état magique associé à la porte  $T$  est l'état  $|\pi/4\rangle = (|0\rangle + e^{i\pi/4}|1\rangle)/\sqrt{2}$ . L'injection, illustrée à

---

2. Les codes stabilisateurs sont des codes de corrections d'erreur quantique. Soient  $n$  opérateurs de Pauli indépendants formant un ensemble complet d'observables qui commutent. Un sous-ensemble de  $n - k$  opérateurs spécifie un sous-espace de dimension  $2^k$ , c'est-à-dire un sous-espace à  $k$  qubits. Ces  $n - k$  opérateurs sont les stabilisateurs qui définissent le code.

3. Les portes sont dites *transversales* si une erreur (sur un qubit physique) ne se propage pas à plus d'une erreur par bloc après l'opération. Une erreur corrigible reste corrigible après l'application d'une porte transversale.

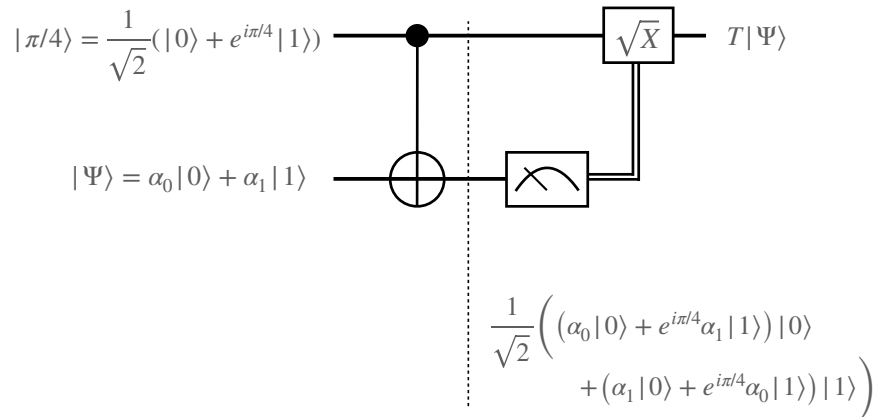


FIGURE 1.2 – Circuit de l’injection équivalent à la porte  $T$ . L’état avant la mesure (ligne pointillée) est donné sous le circuit.

la Fig. 1.2, consiste à intriquer le système avec l’état magique, puis à faire une mesure suivie d’une correction si nécessaire. Obtenir l’état magique peu bruyant est une tâche coûteuse réalisée par la distillation. Ce procédé prend plusieurs états magiques bruyants pour en faire un qui le soit moins. En itérant le processus, on peut atteindre une erreur arbitrairement petite. Pour une précision  $\epsilon$ , la distillation ajoute un coût en  $\mathcal{O}(\log^\gamma(1/\epsilon))$ , où  $\gamma = \log_2\left(\frac{3k+8}{k}\right)$  et  $k$  est le nombre d’états magiques distillés obtenus après le protocole [14]. En somme, ce niveau ajoute son lot de surcoûts. Simplement pour la conversion physique-logique tolérante aux fautes, ce niveau augmente considérablement le nombre de qubits et le nombre d’opérations. De plus, chaque opération logique doit être suivie d’un cycle de correction d’erreur (détection et correction) qui accroît aussi le nombre de qubits et d’opérations, en plus d’un calcul (classique) pour décoder l’erreur la plus probable.

Le **quatrième niveau** est un compilateur au niveau logique. Il traduit un circuit quantique quelconque en une séquence de portes selon un ensemble universel de portes logiques tolérantes aux fautes accessibles à notre appareil. Cet ensemble peut être déterminé notamment par le code de correction d’erreur utilisé au troisième niveau et par les portes natives du système physique. Il existe des algorithmes efficaces qui permettent la décomposition de n’importe quel unitaire en une série de

portes Clifford+ $T$ , et ce, de manière à réaliser l'unitaire à une précision  $\varepsilon$  avec un nombre minimal de portes  $T$ , soit environ  $3 \log_2(1/\varepsilon)$  [15]. Évidemment, la traduction d'une porte unitaire arbitraire dans un ensemble discret de portes ajoute un coût. Aussi, plus le circuit est long, plus on doit être résistant au bruit (nécessite des codes de corrections d'erreur plus coûteux). Il convient donc de réduire autant que possible le nombre de portes à effectuer – particulièrement les portes non transversales – par une optimisation de circuit. Qui plus est, on doit tenir compte de la connectivité des qubits. Est-il possible de réaliser une porte à deux qubits sur toutes les paires? Si ce n'est pas le cas, il faut considérer l'addition de portes SWAP (échange l'information entre deux qubits) pour rendre voisins les qubits qui ne l'étaient pas ou bien l'ajout d'un bus quantique, c'est-à-dire des qubits ancillaires connectant l'entièreté des qubits de données. Différents registres d'optimisations peuvent être faits dans les deux cas pour réduire le coût de cet ajout : position des qubits logiques et physiques, parallélisation de portes commutantes, etc.

Le **cinquième niveau** concerne les algorithmes quantiques et leurs applications. On peut voir les algorithmes comme un ensemble de sous-routines, un paquetage permettant de réaliser ou construire les fonctions nécessaires à la résolution d'une classe de problèmes. On doit pouvoir exploiter les principes de superposition et d'intrication de manière à avoir un avantage sur le classique.

Le **sixième niveau**, le niveau le plus élevé, est simplement l'interface utilisateur. Celle-ci permet de communiquer la séquence de commandes ou les paramètres d'entrées pour une instance d'une classe de problèmes et de lancer le calculateur. Pour pouvoir comprendre le coût réel d'un calcul et s'assurer d'avoir un avantage une fois le tout pris en compte, il faut d'abord comprendre quels types de problèmes un ordinateur quantique peut résoudre de manière efficace, ou au moins, en offrant un gain par rapport à sa contrepartie classique. Autrement dit, il faut comprendre la complexité algorithmiques des problèmes étudiés.

Les travaux de recherches de cette thèse se situent principalement au cinquième niveau, au niveau des algorithmes quantiques et de leurs applications. Par contre, comme chaque niveau ajoute bon nombre de complications et de surcoûts, il faut avoir en tête le portrait global afin d'obtenir des résultats qui soient intéressants. Il faut comprendre à quelle taille de problème un algorithme quantique surpassera les

performances classiques, non seulement en termes d'opérations, mais également de temps réel. En effet, le temps d'exécution d'une porte logique quantique est considérablement plus long que son homologue classique. En outre, l'addition de qubits peut augmenter le bruit, la diaphonie (de l'anglais « crosstalk ») et la complexité de contrôle. Si l'on ne s'attarde qu'à la complexité d'un algorithme, on oublie le détail qui en pratique pourrait faire la différence entre une application (ou instance) faisable avec les ressources à notre disposition et une qui soit simplement hors d'atteinte.

Notons que nous nous sommes consacrés aux ordinateurs quantiques basés sur un modèle en circuit plutôt qu'un modèle analogique. Bien que les deux modèles soient équivalents d'un point de vue algorithmique à un facteur polynomial près [16], nous croyons que pour un usage universel, un calculateur basé sur les circuits a un potentiel légèrement supérieur. Nous ne remettons pas en cause la possibilité d'avoir un calculateur analogique plus performant pour certaines tâches spécifiques [17]. Par contre, le modèle en circuit permet notamment l'usage de la correction d'erreur (offrant un calcul tolérant aux fautes potentiellement plus long) et l'usage de procédures *non physiques* pouvant offrir un gain intéressant (par exemple la méthode de qubitisation, voir chap 2). Notons également qu'un ralentissement polynomial peut causer la perte de tout avantage attendu par rapport au calcul classique.

## 1.2 Accélération quantique

Il existe différents types d'accélération quantique, certains plus concrets, voire indéniables, d'autres plutôt circonstanciels. Dans [18], les auteurs font une synthèse de différentes définitions de l'accélération. On y retrouve notamment comment mesurer ces gains quantiques et aussi comment repérer des situations cachant ou donnant de fausses accélérations. Ces accélérations quantiques sont exprimées en fonction de la taille du problème  $n$ . Elles sont définies simplement par le ratio entre les temps nécessaires à la résolution d'un problème pour l'ordinateur classique et quantique. Notons qu'on ne s'intéresse qu'à l'ordre de grandeur dominant : polynomial, exponentiel, etc. Évidemment, il y a quelques ambiguïtés au niveau notamment des algorithmes comparés. C'est pourquoi il est nécessaire de faire une distinction entre les différentes méthodes de comparaison. Pour de plus amples détails, consulter le tableau 1.1.

Accélération quantique	Définition	Exemple
Démontrable	Existence d'une preuve qu'aucun algorithme classique ne peut surpasser l'algorithme quantique en question	Algorithme de Grover
Simple	Accélération par rapport au meilleur algorithme classique connu	Algorithme de factorisation (Shor)
Potentielle	Accélération par rapport à un ou un ensemble d'algorithmes classiques spécifiés	Simulation de l'évolution d'un système quantique
Limitée	Accélération par rapport à un algorithme classique correspondant à la même approche algorithmique	Recuit quantique en comparaison avec le recuit quantique simulé

TABLEAU 1.1 – Définition des cinq grandes catégories d'accélération quantiques selon [18]

Cette thèse se consacre à l'étude d'heuristiques basées sur des algorithmes ayant une accélération quadratique démontrable. Bien que l'utilité (particulièrement à court terme) de ce genre de gain soit remise en doute considérant le surcoût d'implémentation lié notamment à la correction d'erreur [19], un usage heuristique pourrait apporter un avantage à moyen et long terme. Par exemple, la majorité des algorithmes offrant des gains quadratiques démontrables sont soit une généralisation, soit un cas particulier de l'algorithme de Grover. Or, cet algorithme s'intéresse aux problèmes non structurés. Il est tout à fait possible, voire même probable, que la structure des problèmes facilite les calculs, particulièrement si l'on utilise un algorithme basé sur une évolution ou le recouvrement entre états, comme c'est le cas pour les algorithmes étudiés dans cette thèse.

## 1.3 Évolution adiabatique

Les trois algorithmes développés dans cette thèse sont tous basés, de près ou de loin, sur une discrétisation de l'algorithme d'évolution adiabatique.

Le principe d'évolution adiabatique peut être utilisé directement en tant que calculateur quantique ou en tant qu'algorithme pour un ordinateur en circuit. Un calculateur adiabatique est une version analogique d'un ordinateur quantique où l'on utilise l'évolution d'un Hamiltonien pour arriver au résultat. L'idée est d'encoder un problème où la solution est l'état fondamental, par exemple, d'un hamiltonien  $H_{\text{final}}$ . En initialisant l'ordinateur dans l'état fondamental d'un Hamiltonien plus simple,  $H_{\text{initial}}$ , le calcul se fait par l'évolution du système sous un Hamiltonien dépendant du temps qui relie ces deux derniers, par exemple :

$$H(t) = \left(1 - \frac{t}{\tau}\right) H_{\text{initial}} + \frac{t}{\tau} H_{\text{final}} \quad (1.2)$$

où  $\tau$  est un paramètre adiabatique définissant la vitesse d'évolution.

Ainsi, les outils de contrôle du calculateur sont en fait des hamiltoniens physiques. À  $t = \tau$ , on mesure l'état du système pour obtenir l'état désiré. Si l'évolution est *suffisamment* lente ( $\tau$  suffisamment grand), le système restera à tout moment dans l'état fondamental du Hamiltonien instantané. Ceci est garanti par le théorème adiabatique. Le théorème met en relation la vitesse d'évolution et le gap instantané du hamiltonien ; plus le gap est petit, plus l'évolution devra être lente pour assurer que le système ne sautera pas à un état excité. Il existe différentes versions du théorème adiabatique. Différentes hypothèses permettent différentes garanties de succès que l'on mesure à l'aide de la fidélité entre l'état obtenu et l'état recherché. Il est important de comprendre que les garanties de performance des théorèmes sont des bornes supérieures sur le temps minimal requis pour atteindre une certaine fidélité [20]. En d'autres mots, la fidélité croît avec le temps d'évolution (le nombre d'itérations). Par contre, avoir un temps court n'implique pas nécessairement une petite fidélité. Ce dernier point indique la possibilité d'utiliser des méthodes similaires de manière heuristique (pour des temps plus courts) dans l'espoir de réduire le coût

moyen d'un algorithme en pratique.

Pour une revue plus complète de ces concepts de calcul adiabatique quantique, voir [20].

**Théorème 1 (Théorème adiabatique [21])** Soient  $H(t) = (1-A(t))H_{initial} + A(t)H_{final}$  un hamiltonien aux vecteurs propres  $|E_k(t)\rangle$  satisfaisant l'équation aux valeurs propres  $H(t)|E_k(t)\rangle = E_k(t)|E_k(t)\rangle$ , où  $A(0) = 0$ ,  $A(\tau) = 1$ ,  $E_k(t)$  est la valeur propre correspondant au  $k$ ème vecteur propre et  $E_0(t) < E_1(t) \leq \dots \leq E_m(t)$ . Soit  $\Delta \leq \Delta(t) = E_1(t) - E_0(t)$  le gap minimal pour  $0 \leq t \leq \tau$ . Si à  $t = 0$  un système est dans l'état fondamental  $|E_0(0)\rangle$ , qu'il évolue sous  $H(t)$  pour un temps  $\tau$  et que

$$\frac{1}{\Delta^2} \max_{0 \leq t \leq \tau} \left| \langle E_1(t) | \frac{dH}{dt} | E_0(t) \rangle \right| \leq \varepsilon \quad (1.3)$$

alors

$$|\langle E_0(\tau) | \psi(\tau) \rangle|^2 \geq 1 - \varepsilon^2 \quad (1.4)$$

où  $|\psi(\tau)\rangle$  est l'état du système au temps  $\tau$ .

Un calculateur adiabatique (analogique) est universel. Il est équivalent, à un coût polynomial près, au calculateur basé sur un modèle en circuit. En effet, il a été démontré que le modèle en circuit peut simuler efficacement le modèle adiabatique [22] et l'inverse, c'est-à-dire que le modèle adiabatique peut simuler de manière efficace le modèle en circuit [16].

Une méthode pour prouver que l'évolution adiabatique peut offrir une accélération quadratique démontrable consiste à formuler l'algorithme de Grover dans le cadre de l'évolution adiabatique. En effet, il est simple de démontrer le gain avec un algorithme en circuit, par contre le facteur polynomial permettant le passage d'un modèle à l'autre pourrait a priori mettre en péril le gain quadratique. Dans les sections suivantes, nous présentons une revue de l'algorithme de Grover et démontrons que le modèle analogique conserve l'accélération du modèle en circuit.



## 1.4 Algorithme de Grover

L'algorithme de Lov Grover [8], proposé en 1996, permet de trouver un élément dans une base de données non structurée. Étant donné une banque de données de taille  $N$ , l'algorithme nécessite un nombre de qubits  $\mathcal{O}(\log_2 N)$  et trouve une solution en un temps  $\mathcal{O}(\sqrt{N})$ . En 1999, Christof Zalka démontre que l'algorithme est en fait optimal ; aucun algorithme quantique ne peut avoir une accélération super-quadratique pour les problèmes de recherche non structurés [23].

L'algorithme utilise un oracle qui indique la solution du problème. Lorsqu'il y a plus d'une solution possible, on considère alors la version généralisée de l'algorithme : l'amplification d'amplitude [24]. Le fonctionnement étant le même, nous utilisons les noms d'amplification d'amplitude et d'algorithme de Grover de manière interchangeable.

En pratique, l'oracle de Grover est un opérateur unitaire diagonal dans la base représentant l'ensemble des données  $\{x_0, \dots, x_{N-1}\}$  ;

$$R_f |x\rangle = (-1)^{f(x)} |x\rangle. \quad (1.5)$$

où

$$f(x) = \begin{cases} 1 & \text{si } x \text{ est solution} \\ 0 & \text{sinon.} \end{cases} \quad (1.6)$$

La condition essentielle pour le fonctionnement efficace de la procédure est que cet oracle fasse bon usage de la superposition d'états. En d'autres mots, l'oracle doit évaluer  $f(x)$  pour chaque état d'une superposition, sans provoquer l'effondrement de la fonction d'onde.

Le choix de la représentation d'un état quelconque dans une base orthonormale est arbitraire. Nous pouvons donc décomposer l'état initial  $|\psi\rangle$  dans la base de l'état succès,  $|\psi\rangle = \alpha |\text{Succès}\rangle + \beta |\text{Succès}^\perp\rangle$ , où  $|\text{Succès}^\perp\rangle$  est un état représentant une superposition de tous les états perpendiculaires à  $|\text{Succès}\rangle$ . Pour simplifier, et sans perte de généralité, imaginons une représentation dans un espace à deux dimensions, où

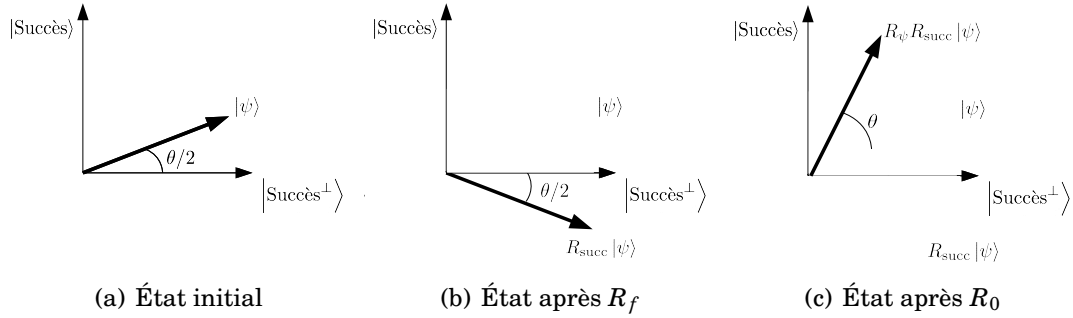


FIGURE 1.3 – Représentation géométrique d'une itération de Grover dans un espace à deux dimensions.

les amplitudes sont réelles ;

$$|\text{Initial}\rangle = \sin \frac{\theta}{2} |\text{Succès}\rangle + \cos \frac{\theta}{2} |\text{Succès}^\perp\rangle.$$

Une représentation géométrique est illustrée à la Fig. 1.3(a). L'état initial est un état typiquement loin de l'état succès (ayant un petit recouvrement avec ce dernier). L'augmentation de la probabilité de succès se fait à l'aide de deux réflexions : une réflexion par rapport à l'état solution (l'état succès) et une autre par rapport à l'état initial, respectivement <sup>4</sup>  $R_f = \mathbb{I} - 2|\text{Succès}\rangle\langle\text{Succès}|$  et  $R_0 = 2|\psi\rangle\langle\psi| - \mathbb{I}$ . L'opérateur  $R_f$  (resp.  $R_0$ ) agit comme réflexion par rapport à l'état succès (resp. l'état initial), en lui ajoutant une phase  $-1$ , voir Fig. 1.3(b) (resp. Fig. 1.3(c)). Cette phase est calculée grâce au marqueur des éléments solutions,  $f(x)$ , tel que décrit à l'équation 1.5.

On représente les  $N$  données à l'aide de  $n = \log(N)$  bits en les identifiant à leur indice dans la base de données. La préparation d'une superposition uniforme sur cet ensemble se fait par l'application d'une tour d'Hadamard sur l'état  $|0\rangle^{\otimes n}$ . Dans ce cas, l'état initial est défini simplement par

$$|\psi\rangle = H^{\otimes n} |0\rangle^{\otimes n} \quad (1.7)$$

$$= \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle \quad (1.8)$$

4. Les représentations peuvent avoir une phase globale  $-1$ . Nous avons choisi cette représentation pour avoir une correspondance plus visuelle pour la Fig. 1.3.

Naturellement, la réflexion autour de cet état est donc

$$\begin{aligned} R_0 &= 2|\psi\rangle\langle\psi| - \mathbb{1} \\ &= H^{\otimes n}(2|00\dots 0\rangle\langle 00\dots 0| - \mathbb{1})H^{\otimes n}. \end{aligned}$$

Notons que la tour d'Hadamard pour la préparation de l'état initial peut être remplacée par n'importe quelle sous-routine unitaire  $U$  (et son inverse  $U^\dagger$ ). Il est donc possible de procéder à l'amplification d'amplitude sur un état de plus grand recouvrement, ou simplement sur un état non uniformément distribué.

L'algorithme de Grover procède donc ainsi :

1. Préparation d'une superposition d'états de l'ensemble des données
2. Répétition des unitaires suivants  $\mathcal{O}(\sqrt{N})$  fois :
  - (a) Application de l'oracle qui inverse la phase des états solutions
  - (b) Application d'une réflexion autour de la superposition initiale
3. Mesurer l'état résultant et répéter toute la procédure jusqu'à obtention de  $f(x) = 1$ .

Le circuit est illustré à la Fig. 1.4. Ainsi à chaque itération de l'étape 2, l'amplitude de l'état succès augmente d'un angle correspondant à  $\theta$  ;

$$\begin{aligned} |\psi\rangle &= \sin\left(\frac{\theta}{2}\right)|\text{Succès}\rangle + \cos\left(\frac{\theta}{2}\right)|\text{Succès}^\perp\rangle \\ \xrightarrow{R_f} &-\sin\left(\frac{\theta}{2}\right)|\text{Succès}\rangle + \cos\left(\frac{\theta}{2}\right)|\text{Succès}^\perp\rangle = \cos(\theta)|\psi\rangle - \sin(\theta)|\psi^\perp\rangle \\ \xrightarrow{R_0} &-\cos(\theta)|\psi\rangle - \sin(\theta)|\psi^\perp\rangle = -\sin\left(\frac{3\theta}{2}\right)|\text{Succès}\rangle - \cos\left(\frac{3\theta}{2}\right)|\text{Succès}^\perp\rangle \\ \dots & (r-1) \text{ itérations} \\ &= \sin\left(\theta\left(\frac{1}{2} + r\right)\right)|\text{Succès}\rangle + \cos\left(\theta\left(\frac{1}{2} + r\right)\right)|\text{Succès}^\perp\rangle \end{aligned}$$

Pour maximiser la probabilité de succès, on choisit idéalement

$$r = \left\lceil \frac{\pi}{2\theta} - \frac{1}{2} \right\rceil, \quad (1.9)$$

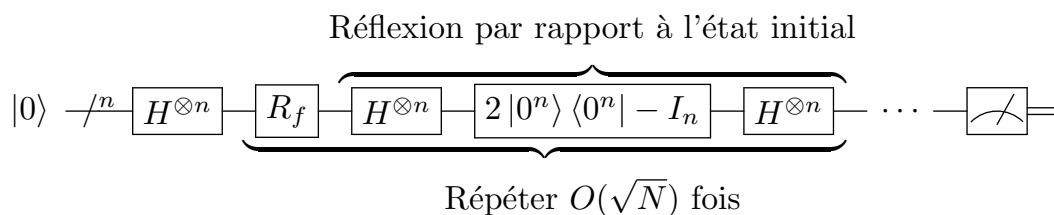


FIGURE 1.4 – Circuit de l’algorithme de Grover

et donc, si  $\sin\left(\frac{\theta}{2}\right) = \frac{1}{\sqrt{N}}$ ,  $r \approx \frac{\pi\sqrt{N}}{4}$ . Notons qu’il s’agit d’une fonction cyclique. L’équation 1.9 dénote donc le nombre d’itérations minimal offrant une mesure optimale. Il est possible de réduire en moyenne le nombre d’appels à l’oracle, en effectuant une mesure avant d’avoir le recouvrement maximal. Le risque de devoir recommencer à zéro sera plus grand, mais le nombre d’appels à l’oracle peut ainsi être réduit d’environ 12,14% par rapport au résultat ci-dessus [23]. Notons que cette réduction ne change pas la complexité qui reste en  $\mathcal{O}(\sqrt{N})$ .

On constate que dans le cas classique, on ne peut pas faire la recherche d’un élément avec une complexité en deçà de  $\mathcal{O}(N)$ . En effet, il faudra en moyenne  $N/2$  appels à l’oracle et  $N-1$  appels dans le pire cas. Intuitivement, l’accélération quantique vient du fait qu’on travaille avec les amplitudes, plutôt que les probabilités. Elle s’interprète également d’un point de vue d’exploration de l’espace des solutions. Le fait de pouvoir évaluer la fonction  $f$  sur l’entièreté du domaine, de la liste, en superposition en un seul appel à l’oracle, nous permet en quelque sorte de pouvoir *classer* notre liste selon une densité de probabilité piquée sur les solutions du problème.

## 1.5 Algorithme adiabatique de Grover

Le calcul adiabatique, étant universel, peut ainsi exécuter tout algorithme en circuit à un surcoût, au pire, polynomial. Il est donc d’intérêt de confirmer si un gain quadratique dans le modèle en circuit – comme celui de Grover – sera perdu ou non lors d’un calcul analogique.

Dans le cadre de l'algorithme adiabatique, définissons les hamiltoniens initial et final du problème décrit par Grover comme étant

$$H_{\text{initial}} = \mathbf{I} - |\psi\rangle\langle\psi|, \text{ et} \quad (1.10)$$

$$H_{\text{final}} = \mathbf{I} - |\text{Succès}\rangle\langle\text{Succès}|. \quad (1.11)$$

Nous pouvons réécrire le hamiltonien dépendant du temps dans la base  $\{|\text{Succès}\rangle, |\text{Succès}^\perp\rangle\}$ , nous permettant de calculer le gap :

$$\Delta(s) = \sqrt{2s^2 \cos(2\theta) + 2s^2 - 2s \cos(2\theta) - 2s + 1}, \quad (1.12)$$

où, pour alléger la notation, nous avons fait le changement de variable  $s = t/\tau$ . Le gap minimal, se trouvant à  $s = 1/2$ , est de  $|\sin(\theta)|$ . On constate que, selon le théorème adiabatique, pour avoir une grande probabilité de succès, il faut un temps  $\tau \gg \frac{3}{\Delta_{\text{min}}^2}$ , perdant ainsi tout gain. Or, ce développement considère un chemin de vitesse constante (interpolation linéaire entre  $H_{\text{initial}}$  et  $H_{\text{final}}$ ), alors que le gap est petit seulement pour  $s = 1/2$ . En utilisant un chemin plus rapide, mais qui ralentit près du gap minimal, nous pouvons obtenir une amélioration substantielle [21,25]. En choisissant un chemin astucieux  $A(t) \neq t/\tau$ , de sorte que  $A(0) = 0, A(\tau) = 1$  et  $\partial_t A = \varepsilon \Delta^2[A(t)]$  [21, 26], nous pouvons retrouver le même gain que le modèle en circuit. En effet, en intégrant  $\partial_t A = \varepsilon \Delta^2[A(t)]$  sur  $t$  et en isolant pour  $\tau$ , nous obtenons

$$\tau = \frac{1}{\varepsilon} \csc(\theta) \sec(\theta) \arctan(\cot(\theta)) \xrightarrow{N \gg 1} \frac{\pi}{2\varepsilon |\sin(\theta)|}, \quad (1.13)$$

Ainsi, nous constatons l'impact important de la vitesse d'évolution, mais également du chemin choisi. Comme on le voit pour Grover adiabatique, un chemin à vitesse constante n'accorde pas d'accélération par rapport au chemin classique, alors qu'un chemin plus rapide là où le gap est plus grand (et plus court là où il se referme), permet de conserver le gain démontré pour l'algorithme en circuit. Il est également possible de retrouver la preuve d'optimalité de l'algorithme dans le contexte adiabatique, pour plus d'information, consulter [21].

Certes, l'évolution adiabatique analogique a le potentiel d'offrir un gain quadratique par rapport à l'algorithme classique, mais qu'en est-il d'une discrétisation de l'algorithme? Comme mentionné précédemment, les modèles en circuit et analogique

sont équivalents à un facteur polynomial près. Bien qu'il puisse y avoir différents facteurs pour différents algorithmes, et qu'il convient de choisir une méthode astucieuse pour traduire un algorithme d'un système à un autre, il est tout de même justifié de vérifier si, de manière générale, le facteur polynomial lié à la discrétisation fait perdre une accélération polynomiale.

La méthode usuelle de passage du modèle adiabatique au modèle en circuit est basée principalement sur une discrétisation du chemin adiabatique suivi par une trotterisation. Par contre, dans cette thèse, nous nous intéressons à une approche fondée sur l'effet Zénon quantique.

## 1.6 Effet Zénon quantique

L'effet Zénon tient son nom du philosophe présocratique Zénon d'Élée, connu pour ses paradoxes voulant démontrer que le mouvement est impossible et donc que nos sens sont trompeurs. Ses paradoxes sont fondés principalement sur le concept d'infinité. Par exemple, le paradoxe de la flèche – probablement celui s'apparentant le plus à l'effet quantique en question – est une expérience de pensée sur une flèche en vol. La prémisse implicite est qu'une somme infinie d'éléments nuls soit zéro. L'idée est que le temps est une succession d'instantanés où la flèche est immobile. En somme, le mouvement devrait donc être impossible. Évidemment, le calcul infinitésimal démontre le non fondé du paradoxe ; la somme d'une infinité d'*instants* résulte bien à une valeur finie et non nulle.

L'effet Zénon quantique est le précepte selon lequel nous pouvons *arrêter* le mouvement d'un système grâce au principe d'effondrement de la fonction d'onde. L'état exact est connu au moment de la mesure. Un peu dans l'idée du paradoxe de la flèche, si à chaque instant on observe le système, aucune évolution n'est possible. En appliquant une série de mesures projectives identiques (selon la même base) et suffisamment rapprochées les unes des autres, nous *empêchons* le système d'évoluer dans le temps. Plus les mesures sont fréquentes, moins le système subit de changement, et donc, plus le recouvrement est grand entre les états obtenus après deux mesures consécutives. Nous pouvons ainsi conserver un état avec une grande probabilité, soit *arrêter le*

*mouvement*, figer l'état dans le temps.

Ici, nous nous intéressons à une variante de l'effet Zénon quantique. Plutôt que d'empêcher le système d'évoluer, nous guidons l'évolution dans la direction désirée. En résumé, l'idée est d'approcher une évolution adiabatique par un effet stroboscopique. Le chemin adiabatique est décomposé en une séquence de mesures définies par les hamiltoniens instantanés. Pour assurer une grande probabilité de succès, l'état d'intérêt aux instants  $i$  et  $i + 1$  doivent avoir un grand recouvrement. Ces états sont respectivement états propres des mesures  $M_i$ , associée au hamiltonien instantané  $H_i$ , et  $M_{i+1}$ , associée à  $H_{i+1}$ .

**Définition : Mesure projective pour un état [27]**

Une opération de mesure projective sur  $|\psi(l)\rangle$  est une opération quantique de la forme :

$$M_l(\rho) = P_l \rho P_l + \epsilon((\mathbb{1} - P_l)\rho(\mathbb{1} - P_l))$$

où  $P_l = |\psi(l)\rangle\langle\psi(l)|$  et  $\epsilon$  est une opération quantique arbitraire qui peut varier selon  $l$ .

**Lemme : Effet Zénon [27]**

Soient  $\{|\psi(l)\rangle\}_{l \in [0, L]}$  un chemin continu d'états et une valeur  $d$  donnée, tel que  $\forall \delta$  :

$$|\langle\psi(l)|\psi(l + \delta)\rangle|^2 \geq 1 - d^2 \delta^2$$

Alors l'état  $|\psi(L)\rangle$  peut être préparé à partir de  $|\psi(0)\rangle$  avec une fidélité  $p > 0$  par  $\frac{L^2 d^2}{1-p}$  mesures projectives intermédiaires.

Plutôt qu'une évolution adiabatique standard, la version discrète de l'algorithme effectue une série de mesures projectives sur les états propres instantanés du hamiltonien avec, par exemple, l'algorithme d'estimation de phase (pour une revue, voir [A.1](#)). On doit donc choisir une discrétisation du chemin du hamiltonien. Le potentiel gain

offert par l'effet Zénon quantique est fondé sur la relation entre la diminution de l'erreur et le raffinement de la discrétisation ; pour un petit déplacement  $\delta$ , la probabilité de projetée  $|\psi(s)\rangle$  sur  $|\psi(s+\delta)\rangle$  diminue avec  $\delta^2$  alors que la distance est linéaire en  $\delta$ . Ainsi, pour un chemin d'états  $\{|\psi(l)\rangle\}$ , l'état final  $|\psi(1)\rangle$  peut être préparé avec une grande fidélité à partir de  $|\psi(0)\rangle$  par l'entremise d'une séquence de mesures projectives sur les états intermédiaires  $|\psi(s_1)\rangle, \dots, |\psi(s_q)\rangle$ , pour  $0 < s_1 < \dots < s_q = 1$  [27]. On observe que

$$\begin{aligned} \prod_i |\langle \Psi(s_i) | \Psi(s_{i+1}) \rangle|^2 &\geq (1 - d^2 \delta^2)^{\lceil 1/\delta \rceil} \\ &\geq 1 - d\delta \end{aligned}$$

En fait, le problème se reformule encore plus clairement en fonction du gap.

**Lemme : Distance des états fondamentaux [28]**

Soient  $H_i, H_f$  deux hamiltoniens tels que  $\|H_f - H_i\| \leq \eta$  et dont le gap est borné inférieurement  $\Delta_i, \Delta_f \geq \Delta$ . Alors,

$$\left| \langle \psi_i^0 | \psi_f^0 \rangle \right| \geq 1 - \frac{4\eta^2}{\Delta^2}$$

où  $|\psi_i^0\rangle$  et  $|\psi_f^0\rangle$  sont respectivement les états fondamentaux de  $H_i$  et  $H_f$ .

En prenant une discrétisation plus raffinée, permettant de borner chaque intervalle  $\|H_{i+1} - H_i\| \leq \frac{\eta}{R}$ , où  $R$  est le nombre de mesures projectives, le lemme ci-dessus nous permet de reformuler la probabilité de succès de chaque projection  $\left| \langle \psi_i^0 | \psi_f^0 \rangle \right|^2 \geq \left(1 - \frac{4\eta^2}{R^2 \Delta^2}\right)^2$  [28]. En choisissant  $R \geq \frac{2\eta}{\Delta} + 1$ , nous obtenons

$$\begin{aligned} \prod_i |\langle \Psi(s_i) | \Psi(s_{i+1}) \rangle|^2 &\geq \left(1 - \frac{4\eta^2}{R^2 \Delta^2}\right)^{2R} \\ &\geq 1 - \frac{8\eta^2}{\Delta^2}. \end{aligned}$$



Une mesure projective sur les états propres des hamiltoniens instantanés doit différencier l'état d'intérêt, soit l'état fondamental, des autres états. Ainsi, une telle mesure doit avoir une précision plus grande que la différence d'énergie entre le premier état excité et l'état fondamental, c'est-à-dire le gap. En prenant en considération le nombre de répétitions moyen pour atteindre la solution, soit l'inverse de la probabilité de succès, et le coût par répétition (en supposant des mesures par estimation de phase), le coût moyen de l'algorithme se révèle être

$$\frac{\text{Coût}}{p_s} \leq \frac{1}{p_s} \sum_R \frac{1}{\Delta} \quad (1.14)$$

$$\leq \frac{2\eta + \Delta}{\Delta^2 - 8\eta^2}. \quad (1.15)$$

En somme, la discrétisation de l'algorithme adiabatique devrait offrir le même potentiel que la version analogique, soit une complexité en  $\mathcal{O}\left(\frac{1}{\Delta^2}\right)$ .

## 1.7 Récapitulatif et présentation des projets

Le but de la thèse est de présenter des algorithmes qui soient optimisés pour des ordinateurs quantiques tolérants aux fautes. Comme la conception de ces appareils est ardue, nous devons réduire autant que possible les ressources nécessaires (qubits et opérations) particulièrement si nous voulons un gain réel sur l'algorithme classique à moyen terme. Les ressources les plus coûteuses sont les portes non transversales représentées dans cette thèse par la porte T ou plus généralement comme les portes du troisième niveau de la hiérarchie de Clifford.

Les algorithmes étudiés sont tous des heuristiques basés sur une discrétisation de l'évolution adiabatique. La discrétisation en elle-même peut s'apparenter à un choix de chemin, linéaire ou non. Un choix astucieux peut permettre de meilleures performances. C'est entre autres une des approches utilisées pour le premier algorithme présenté au chapitre suivant : un algorithme de préparation d'état par une évolution adiabatique discrète utilisant l'effet Zénon. Nous étudions également l'impact de différentes techniques sur le coût total de l'algorithme.

Le second algorithme est une version quantique des marches aléatoires. Ces dernières sont très similaires à l'algorithme de Grover. Combiné à l'évolution adiabatique, cet algorithme offre un gain potentiellement quadratique pour la préparation d'une distribution stationnaire. Toujours dans l'optique d'optimiser les ressources, nous présentons un usage heuristique et comparons sa performance à son homologue classique.

Le dernier algorithme présenté est semblable au premier, à l'exception que les mesures projectives sont remplacées par des réflexions. En ce sens, cet algorithme est très ressemblant à Grover. Nous avons évalué ses performances pour un type de problème NP-difficile, soit MAX 2-SAT. Par contre, rien n'indique qu'il ne pourrait pas également être utilisé pour de la préparation d'état.

## Chapitre 2

# Préparation d'état par évolution adiabatique discrète

*Hilbert space is gratuitously big*

— Caves and Fuchs, 1996 [29]

L'une des méthodes pour laquelle le gain attendu est le plus grand est la simulation d'un système quantique dont la dynamique est décrite par un hamiltonien. On cherche alors à modéliser l'évolution temporelle d'un tel système pour un état initial donné. Contrairement aux cas bosoniques, pouvant généralement être étudiés par du Monte-Carlo quantique par exemple, la simulation de systèmes fermioniques est particulièrement difficile, ou non-efficace, à cause du problème de signe [30]. L'antisymétrie des fonctions d'onde fermioniques implique qu'à chaque permutation de deux particules, une phase  $-1$  apparaît. La complexité d'une simulation augmente alors non seulement avec la taille du système et le nombre d'orbitales ou d'atomes, mais aussi avec le nombre d'arrangements possibles des fermions.

Pour certaines méthodes de simulation, il peut être requis d'initialiser le système dans un état d'intérêt, par exemple dans son état fondamental. Il faut donc traiter le problème statique d'abord, soit l'équation de Schrödinger indépendante du temps, avant de pouvoir résoudre la dynamique. Or, cette initialisation n'est pas triviale pour l'ordinateur quantique ; elle est en fait plus naturelle pour le calcul classique. Dans l'article présenté dans ce chapitre, nous analysons le coût associé à une telle

préparation par un algorithme d'évolution adiabatique discrète. L'idée est en quelque sorte de reformuler le problème statique en un problème dynamique, un problème d'évolution.

Puisque trouver l'état fondamental d'un hamiltonien est en soit un problème statique, les méthodes classiques peuvent être utilisées pour résoudre cette première partie du problème et ainsi nous aider à calculer le coût associé à la préparation d'état sur un appareil quantique. À noter que connaître l'état fondamental du système n'indique pas comment préparer un tel état. Les méthodes classiques peuvent donc, d'une part, calculer les ressources nécessaires à la préparation d'état et, d'autre part, assister la procédure quantique en optimisant certains paramètres. Toutefois, elles ne remplacent pas l'initialisation sur l'ordinateur quantique, puisqu'elles ne permettent pas de résoudre la dynamique du problème.

Nous avons choisi de faire l'étude de notre algorithme appliqué au modèle de Hubbard [31]. Celui-ci est d'un grand intérêt en physique de l'état solide et est difficile à résoudre. En effet, il n'existe aucune solution analytique du modèle en plus d'une dimension. Pour une revue du modèle de Hubbard, voir A.2. Afin de pouvoir étudier des réseaux 2D, nous avons utilisé des méthodes approximatives pour calculer les quantités d'intérêt comme l'état fondamental et le gap à plusieurs particules d'un hamiltonien instantané. La technique choisie est une méthode variationnelle de réseaux de tenseurs : la DMRG [32]. Pour une revue des réseaux de tenseurs, voir [33].



L'article combine et compare différentes méthodes afin de réduire au maximum les ressources nécessaires à la préparation d'état. Parmi ces techniques, on compte l'estimation de phase (pour une revue, voir A.1), une méthode d'optimisation des pas, une méthode de rembobinage [34] ainsi que la qubitisation [35].

## 2.1 Article

L'idée originale est du professeur David Poulin, principal superviseur du projet. Nous avons défini le projet ensemble puis j'ai fait les simulations, calculs et démonstrations nécessaires aux résultats qui ont été certifiés en partie par chacun des co-auteurs. Les co-auteurs et moi-même avons tous participé aux discussions techniques du projet ainsi qu'à l'écriture de l'article.

Cet article a été publié au journal *Physical Review A* :  
Jessica Lemieux, Guillaume Duclos-Cianci, David Sénéchal et David Poulin. Resource estimate for quantum many-body ground-state preparation on a quantum computer. *Physical Review A*, 103(5), 052408 (2021), doi :[10.1103/PhysRevA.103.052408](https://doi.org/10.1103/PhysRevA.103.052408).

## Resource estimate for quantum many-body ground-state preparation on a quantum computer

Jessica Lemieux <sup>1</sup>, Guillaume Duclos-Cianci,<sup>1</sup> David Sénéchal <sup>1</sup>, and David Poulin<sup>1,2,3</sup>

<sup>1</sup>*Département de Physique & Institut Quantique, Université de Sherbrooke, Québec, Canada J1K 2R1*

<sup>2</sup>*Canadian Institute for Advanced Research, Toronto, Ontario, Canada M5G 1Z8*

<sup>3</sup>*Quantum Architecture and Computation Group, Microsoft Research, Redmond, Washington 98052, USA*



(Received 15 June 2020; revised 23 February 2021; accepted 8 March 2021; published 10 May 2021)

We estimate the resources required to prepare the ground state of a quantum many-body system on a quantum computer of intermediate size. This estimate is made possible using a combination of quantum many-body methods and analytic upper bounds. Our routine can also be used to optimize certain design parameters for specific problem instances. Lastly, we propose and benchmark an improved quantum state preparation procedure. We find that it reduces the circuit  $T$ -depth by a factor as large as  $10^6$  for intermediate-size lattices.

DOI: [10.1103/PhysRevA.103.052408](https://doi.org/10.1103/PhysRevA.103.052408)

### I. INTRODUCTION

Quantum many-body systems are notoriously difficult to simulate on classical computers. This observation led Feynman to suggest that they could be efficiently simulated using quantum computers instead [1]. The *dynamics* of quantum systems are dictated by Schrödinger's equation, which results in a unitary time evolution. Beginning with Lloyd [2] and Aharonov and Ta-Shma [3], the design of increasingly efficient quantum circuits solving Schrödinger's equation has been an intensive area of research in the past decade (see, e.g., Refs. [4–15]).

Beyond the dynamics, the simulation of a quantum system also requires setting initial conditions. Of particular interest are properties at low temperature, where, to a good approximation, the system is initially in its ground state. Thus, to solve the *static* problem, a quantum circuit must be constructed that maps a fiducial initial state to the ground state of the system of interest. This problem is generally QMA-complete [16–19], so the existence of a general-purpose efficient procedure is believed to be impossible. Nonetheless, heuristic methods have been proposed [3,20–23] that could be efficient for specific physical systems, and exponential-time algorithms [24,25] could be sufficiently fast for intermediate-size problems.

The situation is somewhat reversed when it comes to classical computers. There, a host of methods have been devised to approximately solve the static problem. These include density-functional theory, quantum Monte Carlo, and tensor-network methods, to name a few (see, e.g., Refs. [26–28] for reviews). However, the growth of entanglement in time typically makes it impossible to classically simulate the dynamics of quantum many-body systems.

The goal of this article is to compare and optimize various approaches to provide an estimate of the resources (number of gates) required to solve a quantum simulation problem on an intermediate-size quantum computer, say of a few hundred qubits. For the sake of concreteness, we study the Hubbard

model, the paradigmatic model of strongly correlated electrons. We will also focus on a specific, generally nonefficient algorithm to prepare the ground state of this model; it is a discrete version [20] of the quantum adiabatic state preparation [3,29,30] with basic components that have also been presented in Ref. [31]. For simplicity, we consider a linear interpolation with an optimized schedule. The latter provides a rough upper bound that can surely be improved, e.g., by adding a symmetry-breaking field to lift gapless modes [32]. The runtime of this algorithm depends on properties of the system usually unknown analytically. We employ tensor-network methods to (approximately) solve the static problem on a classical computer. Aside being useful to benchmark the quantum algorithm, this classical side computation could also be used to optimize certain parameters of the adiabatic state preparation algorithm, such as deciding on an interpolation schedule. This idea of combining classical and quantum algorithm is often seen in Quantum Approximate Optimization Algorithm (QAOA), for example [33].

### II. THE HUBBARD MODEL

The Hubbard model is defined by the Hamiltonian

$$H(T_{\text{hub}}, U) = T_{\text{hub}} \sum_{\sigma} \sum_{\langle i, j \rangle} (c_{i\sigma}^{\dagger} c_{j\sigma} + c_{j\sigma}^{\dagger} c_{i\sigma}) + U \sum_i n_{i\uparrow} n_{i\downarrow}, \quad (1)$$

where the operator  $c_{i\sigma}^{\dagger}$  creates an electron of spin  $\sigma = \{\uparrow, \downarrow\}$  at site  $i$ , its adjoint  $c_{i\sigma}$  is the corresponding annihilation operator, and  $n_{i\sigma} = c_{i\sigma}^{\dagger} c_{i\sigma}$  is the number operator. The notation  $\langle i, j \rangle$  indicates nearest-neighbor sites on a lattice. The constant  $T_{\text{hub}}$  in the kinetic term defines the energy unit and is henceforth set to  $T_{\text{hub}} = 1$ .  $U$  is the strength of the Coulomb interaction, limited to electrons on the same site.

The kinetic term tends to delocalize electrons, and indeed the limit  $U/T_{\text{hub}} = 0$  is easily solved as a free Fermi gas. At the other extreme ( $T_{\text{hub}}/U = 0$ ), the electrons are perfectly localized on each site at half filling. In between

these two extremes, the kinetic and Coulomb energies are in competition and the solution is highly nontrivial. In fact, in the thermodynamic limit, the ground-state phase diagram of the two-dimensional Hubbard model is expected to feature various broken-symmetry states (antiferromagnetic, superconducting, etc.) depending on the electron density, with a vanishing energy gap.

In this work, we set the value of  $U$  to be twice the number of neighbors—e.g.,  $U = 4$  for a one-dimensional (1D) chain,  $U = 6$  for a ladder and  $U = 8$  for a two-dimensional (2D) rectangular lattice—and consider a 10% electron doping above half filling, with an approximately equal number of up and down spins. For small system sizes, we found that this doping has the smallest gap at fixed number of spins. Therefore, with these parameters, the systems should have a small gap along the adiabatic path chosen, which defines hard instances for the state preparation algorithm.

### III. DISCRETE ADIABATIC STATE PREPARATION

The adiabatic algorithm [29,30] leverages a quantum computer's ability to simulate Schrödinger's equation in order to prepare the ground state of a target Hamiltonian  $H(\tau)$  using the adiabatic theorem. This is made possible by beginning with a Hamiltonian  $H(0)$  with a known and easy-to-prepare ground state, and slowly morphing it into the Hamiltonian of interest  $H(\tau)$ . The algorithm is efficient if  $H(t)$  has a gap,  $\Delta(t) = E_1(t) - E_0(t)$ , polynomial in the system size at all times  $0 \leq t \leq \tau$ .

In this work, we will use a discrete version of the adiabatic algorithm [20], reminiscent of the quantum Zeno effect. Recall that the latter employs projective measurements to freeze the unitary evolution of a quantum system. Here, we instead use a sequence of time-dependent measurements to drag the state of a quantum system that is otherwise static. If the change in the measurement basis is small at every step, the outcome is almost deterministic.

Concretely, we choose a discrete sequence of Hamiltonians  $H_j$ ,  $j = 0, 1, \dots, L$  with corresponding ground states  $|\psi_j^0\rangle$ . Given the fidelity  $F_j = |\langle \psi_{j-1}^0 | \psi_j^0 \rangle|^2$  between two consecutive ground states, we can express

$$|\psi_{j-1}^0\rangle = \sqrt{F_j} |\psi_j^0\rangle + \sqrt{1-F_j} |\bar{\psi}_j^0\rangle,$$

where  $|\bar{\psi}_j^0\rangle$  denotes a state orthogonal to the ground state of  $H_j$ . We define a binary projective measurement by the projectors  $Q_j = |\psi_j^0\rangle\langle\psi_j^0|$  and  $\bar{Q}_j = I - Q_j$ . If the system is in state  $|\psi_{j-1}^0\rangle$ , then a measurement of  $\{Q_j, \bar{Q}_j\}$  will produce the outcome  $Q_j$  with probability  $F_j$ , causing the state to collapse into  $|\psi_j^0\rangle$ . Successively measuring for  $j = 1$  to  $L$  will produce the desired state  $|\psi_L^0\rangle$  with a global success probability  $p = \prod_{j=1}^L F_j$ . The main interest of using projective measurements instead of adiabatic evolution is that errors do not accumulate. The error is limited to the one of the last projective measurement.

The binary measurement  $\{Q_j, \bar{Q}_j\}$  is a coarse-grained energy measurement and can in principle be performed from quantum phase estimation [34–36] using the operator  $U_j = e^{-iH_j}$ . Recall that this algorithm is an effective way of estimating the eigenvalues  $e^{i\phi_k}$  of a unitary operator  $U$ . Thus, to

ensure that the quantum phase estimation algorithm differentiates all eigenstates, we normalize  $H_j$  such that its eigenvalues are between 0 and  $2\pi$ . A measurement accuracy  $\epsilon$  requires a simulation time of  $O(1/\epsilon)$ . In the present setting, we are interested in distinguishing the ground state of  $H_j$  from the rest of the spectrum. Based on the above properties of quantum phase estimation, the binary measurement  $\{Q_j, \bar{Q}_j\}$  requires in general simulating the time evolution under  $H_j$  for a time  $t_j = 1/\Delta_j$ , where  $\Delta_j$  is the spectral gap of  $H_j$ . This can be thought of as a manifestation of the Heisenberg time-energy uncertainty relation.

Note that, because nondestructive energy measurements are central to many quantum algorithms, other methods have been devised recently to realize them [37]. Since the relation between the complexities of these methods is relatively well understood, our results can be easily translated to any of these alternative methods by replacing  $t_j$  by the appropriate complexity metric. Whether based on a simulation of time evolution or otherwise, in all cases, a circuit implementation of the binary measurement  $\{Q_j, \bar{Q}_j\}$  requires knowing the ground-state energy  $E_j^0$  and the spectral gap  $\Delta_j$ . We will return to this requirement when describing the simulation method.

In the following, we will estimate the complexity of state preparation in terms of the total simulation time. For a given Hamiltonian sequence  $\{H_j\}_{j=0}^L$ , the duration of a successful state preparation is simply

$$t_{\text{total}} = \sum_{j=1}^L t_j = \sum_{j=1}^L \frac{1}{\Delta_j}. \quad (2)$$

The algorithm succeeds with probability  $p$ . To achieve any desired overall success probability  $1 - \epsilon$ , it needs to be repeated  $\log(\epsilon)/\log(1-p)$  times, resulting in a total time to solution (TTS) [38] of

$$\text{TTS}(\epsilon, \{H_j\}) = \frac{\log(\epsilon)}{\log(1 - \prod_{j=1}^L F_j)} \sum_{j=1}^L \frac{1}{\Delta_j}. \quad (3)$$

The TTS needs to be minimized over some Hamiltonian sequences, achieved by the following procedure: We initialize the sequence with  $\{H_0, H_1\}$  where  $H_0$  is the simple Hamiltonian ( $U = 0$  in our case) and  $H_1$  is the Hamiltonian of interest. At iteration  $L$ , we have the sequence  $\{H_j\}_{j=0}^L$  and the corresponding fidelities  $\{F_j\}_{j=1}^L$ , and for  $k = \text{argmin}_j F_j$ , we perform the assignment

$$H_j \leftarrow H_j \text{ for } j = 0, \dots, k-1, \quad (4)$$

$$H_k \leftarrow (H_{k-1} + H_k)/2, \quad (5)$$

$$H_{j+1} \leftarrow H_j \text{ for } j = k, \dots, L+1. \quad (6)$$

In other words, we add a Hamiltonian halfway between the two Hamiltonians with the lowest fidelity in the sequence. We end the procedure when  $\text{TTS}(\epsilon, \{H_j\})$  has stopped decreasing for a few iterations, i.e., when we are convinced that the algorithm found the minimum. The latter defines the optimal trade-off between space and time cost: Few repetitions of a long high-fidelity iteration or several shorter iterations of lower fidelity.

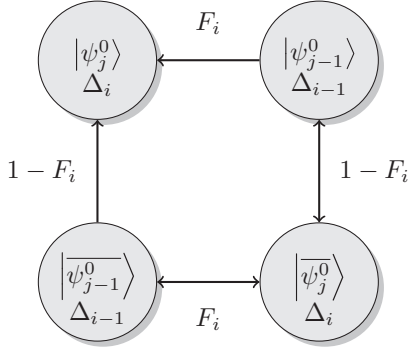


FIG. 1. The cost of the rewind procedure for one step will be the sum of all paths that start at the top-right state and end at the top-left one. The probability to follow an arrow is given by the fidelity  $F$  and its cost is given by the gap  $\Delta$  where the arrow ends.

#### IV. REWIND PROCEDURE

The above procedure has a finite probability of failure, and in case of failure the whole state preparation must be restarted from scratch. Here, we propose a modification to the algorithm which avoids such hard reboots. The technique was introduced in Ref. [39] and used in Refs. [21,32].

At step  $j$ , the register is in state  $|\psi_{j-1}^0\rangle$  and the measurement  $\{Q_j, \bar{Q}_j\}$  is performed. The outcome  $Q_j$  (resp.  $\bar{Q}_j$ ) occurs with probability  $F_j$  (resp.  $1 - F_j$ ) and yields the state  $|\psi_j^0\rangle$  (resp.  $|\psi_j^0\rangle$ ). Instead of rejecting this outcome and rebooting to  $j = 0$ , we alternate measurements of  $\{Q_{j-1}, \bar{Q}_{j-1}\}$  and  $\{Q_j, \bar{Q}_j\}$  and halt whenever  $Q_j$  occurs.

It can easily be shown [39] that the probability of a  $Q_{j-1} \leftrightarrow Q_j$  or a  $\bar{Q}_{j-1} \leftrightarrow \bar{Q}_j$  transition is  $F_j$ , while the other transitions  $Q_{j-1} \leftrightarrow \bar{Q}_j$  and  $\bar{Q}_{j-1} \leftrightarrow Q_j$  have complementary probabilities  $1 - F_j$ . Thus, each step in the Hamiltonian sequence can be seen as a random walk on the graph depicted on Fig. 1, with initial condition  $|\psi_{j-1}^0\rangle$  and absorbing state  $|\psi_j^0\rangle$ . The cost of each transition is given by the inverse of the spectral gap  $\Delta$  of the target state. These considerations lead to a simple geometric series for the average time  $A_j$  to realize step  $j$ :

$$A_j = \frac{F_j}{\Delta_j} + 2 \sum_{k_1=1}^{\infty} \sum_{k_2=0}^{k_1} (1 - F_j)^{2k_1} F_j^{2k_2+1} \times \left( \frac{k_1 + k_2 + 1}{\Delta_j} + \frac{k_1 + k_2}{\Delta_{j-1}} \right). \quad (7)$$

The average of the TTS is  $\text{TTS}(\{H_j\}) = \sum_{j=1}^L A_j$  and can be minimized over Hamiltonian sequences  $\{H_j\}$  following the procedure described in the previous section.

#### V. QUBITIZATION

The preparation cost is a function of the gap and path of evolution, but the complexity of the whole preparation also depends on the implementation of the time-evolution operator. In Ref. [37], a unitary walk operator  $W = \exp\{i \text{acos}(\bar{H}t)\}$  is introduced with an efficient implementation. The spectrum of the Hamiltonian is reversed and renormalized,  $\bar{H}_j = \mathbb{I} - H_j/\mathcal{N}_j = \sum_l |\beta_l^j\rangle^2 P_l$ , where  $\mathcal{N}_j$  is the normalization factor

and the  $P_l$  are Pauli operators. Note that, since we considered the spectrum of  $H_j$  to be between 0 and  $2\pi$ ,  $\mathcal{N}_j = 2\pi$  in the current case. The walk operator maps the Hamiltonian to a higher-dimensional space,  $W = SVe^{i\tau}$  where we defined

$$B|0\rangle = \sum_j \beta_j |j\rangle, \quad (8)$$

$$S = B(1 - 2|0\rangle\langle 0|)B^\dagger, \quad (9)$$

$$V = \sum_j |j\rangle\langle j| P_j \quad (10)$$

as in Refs. [12,37,40].

The algorithm described in the previous sections uses a discretization of the time-dependent Hamiltonian

$$H(t) = \left(1 - \frac{t}{\tau}\right)H_0 + \frac{t}{\tau}H_f, \quad (11)$$

where  $\tau$  is the adiabatic parameter,  $H_0$  is the initial Hamiltonian with an easy-to-prepare ground state, and  $H_f$  is the final Hamiltonian with the target ground state. Recall that energy measurement, such as quantum phase estimation, uses the unitary  $e^{iH_j}$  to decompose the current state in the  $k$  eigenstates of  $H_j$ ,

$$QPE|0\rangle|\psi_{j-1}^0\rangle = \sum_k \langle \psi_j^k | \psi_j^0 \rangle |E_j^k\rangle |\psi_j^k\rangle, \quad (12)$$

the algorithm mimics the evolution by partial energy measurements in order to drag the state towards the ground state of the next instantaneous Hamiltonian.

When adding qubitization, the initial state is the corresponding eigenstate of  $W_0$ ,  $|\varphi_0^0\rangle$ . The eigenstates of  $W_j$  are given by

$$|\varphi_j^k\rangle = \frac{1}{\sqrt{2}} \left( \left[ 1 \mp \frac{i\bar{E}_j^k}{\sqrt{1 - (\bar{E}_j^k)^2}} \right] \sum_l \beta_l^j |l\rangle |\psi_j^k\rangle \pm \frac{i}{\sqrt{1 - (\bar{E}_j^k)^2}} \sum_l \beta_l^j (\mathbb{1} \otimes P_l) |l\rangle |\psi_j^k\rangle \right).$$

The qubitized algorithm uses the same discretization  $\{H_j\}$ , but with the corresponding unitary  $W_j = \exp\{t \text{acos}(\bar{H}_j)\}$  for the partial energy measurement. The states will then go from  $|\varphi_{j-1}^0\rangle$  to  $|\varphi_j^0\rangle$ . Note that it does not correspond to any meaningful physical evolution. Thus, performing phase estimation of  $W_j$  instead of  $e^{iH_j}$  will lead to the preparation of a state equivalent to the ground state of the Hubbard model, up to an isometry. In addition, the implementation of  $W_j$  is exact and, in general, costs less than the implementation of  $e^{iH_j}$ , because  $W_j$  can be implemented with approximately the same amount of resources needed in a single Trotter step [37].

If needed, at the end of the algorithm, the final state is mapped back to the lower-dimensional space with a known isometry. However, some properties such as any static expectation values, can be derived directly from  $|\varphi_f^0\rangle$ , but since  $W$  is not a physical Hamiltonian, we do not expect its dynamic to behave as  $H_f$ .

In the usual adiabatic setting, we assume that this time-dependent Hamiltonian of Eq. (11) has a nonvanishing gap for  $0 \leq t \leq \tau$  and the scaling is bounded by  $O(1/\Delta^2)$  [41].



Intuitively, the scaling can be understood as follows: Each implementation of the QPE needs to differentiate the ground state of the first-excited state, so the required precision of the energy scales with  $1/\Delta$ . The density of steps, i.e., the discretization, depends on how fast the time-dependent Hamiltonian spectrum changes, which can be partially characterized by the gap.

With the qubitization protocol, we have a degeneracy of the states but it will not affect the protocol since both  $k$ th eigenstates of  $W_j$  correspond to eigenstate  $k$  of  $H_j$ . The relation to the gap of interest, the one that differentiates  $k = 0$  from  $k = 1$  is given by

$$\Delta_{W_j} = \text{acos}\left(1 - \frac{\Delta}{\mathcal{N}_j}\right). \quad (13)$$

In some cases, the gap may decrease, which is not good for our purpose. However, for small gaps, in many cases, we will have a significant advantage since

$$\Delta_{W_j} = \text{acos}\left(1 - \frac{\Delta}{2\pi}\right) > \Delta \text{ for } \Delta \lesssim 0.32. \quad (14)$$

To summarize, the discretization should still be bound by  $1/\Delta$ , but the QPE will now scale with  $1/\Delta_{W_j}$ .

For more details on qubitization applied in this context, see Ref. [31].

Qubitization affects the success probability and the gap in an instance-dependent manner. Thus, we cannot formally evaluate its impact on the cost. To assess the efficiency of the algorithm, we performed numerical simulations.

## VI. SIMULATION METHOD

The resources required for a given state preparation depend on the gap of every Hamiltonian in the sequence and the fidelity between the ground state of two consecutive Hamiltonians. For small lattices, these can be computed by Lanczos exact diagonalization [42]. However, this method becomes intractable for system sizes of up to a hundred qubits. To deal with this issue, we use the density-matrix renormalization group (DMRG, see Ref. [43] for a review), a tensor network method that represents quantum states by one-dimensional networks known as matrix product states (MPSs). DMRG is thus the method of choice to study one-dimensional systems numerically, or two-dimensional systems of small width [44]. The method has been implemented using the ITENSOR C++ library [45]. While the results are approximate, we validate the small-size instances with exact diagonalization.

This simulation method is not only of interest for classically benchmarking the quantum state preparation algorithm but, more broadly, can be used to optimize the Hamiltonian sequence. In addition, the circuit implementation of the projective measurement  $\{Q_j, \bar{Q}_j\}$  requires an accurate estimate of the ground-state energy and spectral gap of  $H_j$ . While these could potentially be obtained from the quantum algorithm itself with some extra cost, it is reasonable to assume that, for intermediate size simulations, classical methods will be sufficiently powerful to provide an accurate solution to the static quantum problem.

Details on the simulation method and the parameters used are provided in the Appendix.

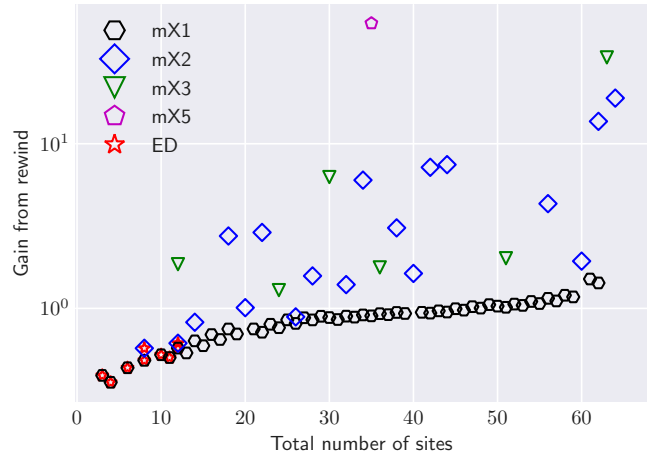


FIG. 2. Gain obtained from the rewind procedure to prepare the ground state of the 10% doped Hubbard model on  $m \times k$  lattices. The units of the TTS are set by the natural units of the Hubbard model ( $T_{\text{hub}} = 1$ ).

## VII. NUMERICAL RESULTS

To obtain a general estimate of the resource scaling, we applied the numerical method outlined above to the Hubbard model on systems of various shapes and sizes, ranging from 2 to 65 sites. Some shapes, such as a square system, have additional ground-state degeneracies because of discrete symmetries. For simplicity, we explicitly discarded such systems. The problem could be circumvented by using a symmetry-breaking field [32], or by using an initial state in a fixed symmetry sector and using a Hamiltonian sequence that preserves this symmetry.

In all cases, the initial Hamiltonian  $H_0$  was obtained by turning off the Coulomb interaction  $U$ . The corresponding ground state is a fermionic Gaussian state which can be easily prepared on a quantum computer [7].

To adiabatically reach the final Hamiltonian, we perform a linear interpolation. It has been shown that this is generally not an optimal path [32], but it has the advantage of simplicity and universality.

The gain is defined as the ratio of the minimum TTS of the first method to the minimum TTS of the second method. The improvement brought by the rewind procedure alone is shown in Fig. 2 and the gain when added to qubitization is shown in Fig. 3. We observe that for small systems, the results obtained with DMRG is in near perfect agreement with exact diagonalization. This leads us to trust that it provides reliable estimates for larger systems.

With some exceptions, the rewind procedure for small gap systems (smaller than 0.1) offers a gain of about one order of magnitude. These systems include chains of about 50–60 sites and 2D systems of 20–60 sites. While our simulations are too limited to extract a clear trend, the improvement appears to increase with lattice size—or, equivalently, with the inverse gap. The rewind procedure will thus become crucial, already for intermediate-size quantum simulation algorithms.

The qubitization introduced in Ref. [37] offers a gain as large as  $10^2$  over the rewind procedure, i.e.,  $10^3$  over the standard discrete adiabatic state preparation algorithm. Since

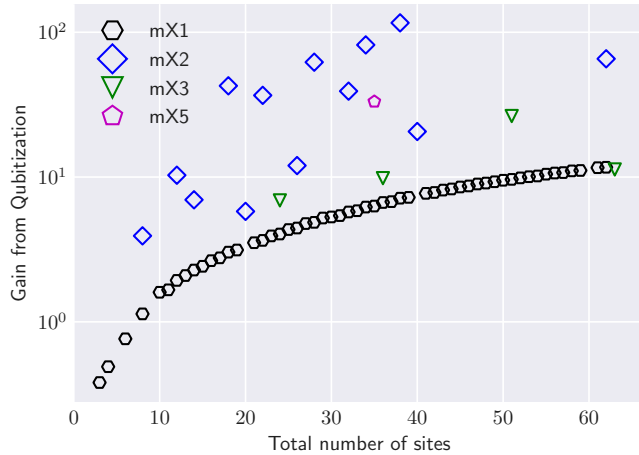


FIG. 3. Gain obtained from qubitization with rewind compared with a direct rewind procedure.

this method can efficiently prepare the eigenstates of its walk operator, it can greatly reduce the circuit depth.

The original adiabatic algorithm is known to have a scaling bounded by  $1/\Delta^2$  [41]. However, in the formulas introduced for our TTS computation, Eqs. (3) and (7), only a factor of  $1/\Delta$  is explicit. Nevertheless, the fidelity term has an implicit dependence on  $\Delta$  related to the density of steps in the discretization. Because we use a TTS approach to define the schedule, we expect the contribution of that term to be at most  $1/\Delta$ , but it is likely faster for each instance. The scaling improvement given by the faster schedule in our current application is studied in the Appendix.

To obtain the circuit complexity, we need to quantify the number of elementary gates required to compute the operator  $U_j = e^{-iH_j}$ . There are different ways to approximate time-evolution operator  $U(t) = e^{-iHt}$  as a quantum circuit, resulting in different gate counts, that have recently been estimated in Ref. [46]. In the most efficient case, the simulation of a 100-qubit system for a time  $t = 100$  requires  $10^9 T$  gates. Since these product-formula algorithms can be parallelized, this implies a  $10^7$  circuit depth. So to produce a time-evolution operator  $U(t)$  with  $t \approx 10^5$ – $10^7$  requires a quantum circuit of depth  $10^{12}$ – $10^{14}$  (ignoring logarithmic corrections). With a microsecond logical gate time, the quantum computation time would be from a week to a year long.

If our ground-state measurement uses the unitary walk operator of Ref. [37] instead of the time-evolution operator, then the spectral gap  $\Delta$  in formulas (3)–(7) should be replaced by  $\arccos(1 - \Delta/N)$ . We find that this method requires  $10^4$ – $10^6$  applications of the unitary walk operator instead of the  $10^5$ – $10^7$  unit-time evolution for systems with  $N \approx 100$  sites. The circuit depth for implementing a single walk operator to precision  $\epsilon$  can be compressed [47] to  $3 \log(N) \log \frac{1}{\epsilon}$ , where  $\epsilon$  is the accuracy with which logical gates are synthesized. An accuracy  $\epsilon = \sqrt{\Delta}/(100N^2)$  should do [37], resulting in a circuit depth  $10^5$ – $10^8$  for  $N \approx 100$ .

These circuit complexities are shown in Fig. 4, which compares the  $T$ -depth of the rewind procedure using product formulas simulated in Ref. [46] with all procedures described

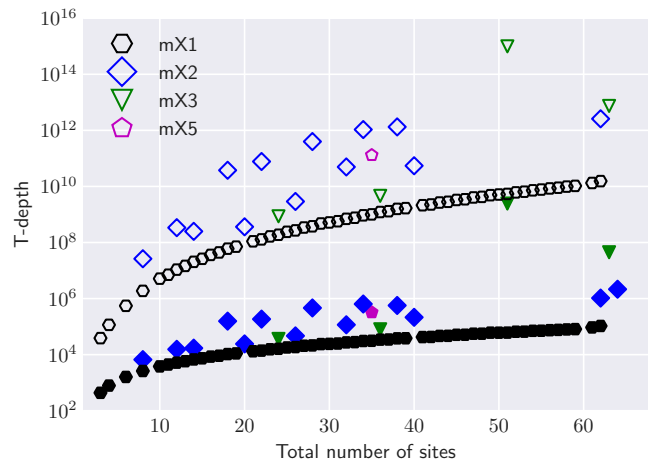


FIG. 4.  $T$  depth of the circuit presented in this work. Discrete adiabatic evolution with qubitization and rewind procedure (filled symbols) vs high-order products using sixth-order formulas with empirical error bound [46] and rewind procedures only (empty symbols).

in this work. For systems of intermediate sizes, the gain is between one and six orders of magnitude.

## VIII. CONCLUSION

Building on well-established quantum many-body methods for classical computers, we have devised a procedure to estimate the cost of quantum ground-state preparation on quantum computers. Besides its use as a benchmark, our procedure could assist in the design and optimization of the quantum simulation algorithm.

Our results show a significant improvement in required resources over the theoretical adiabatic bound. Indeed, qubitization lowers the implementation cost of the unitary used and the precision needed in quantum phase estimation. The latter changes part of the scaling from  $1/\Delta$  to  $1/\arccos(1 - \Delta/N)$ , which corresponds to a gain for the systems studied in this work.

By extrapolating the general trend of our results, we can predict that a time evolution as long as  $10^6$ – $10^{14}$  (in  $T$  depth) is required to prepare the ground state of an intermediate-size Hubbard system with the adiabatic algorithm. We have also proposed an improved adiabatic optimization that decreases these times to the range  $10^4$ – $10^8$ . The overall gain can be as large as  $10^6$  for intermediate-size lattices.

These drops in computing time could be even more impressive when combined to a clever adiabatic path. For example, one could create a nonlinear interpolation, by using a symmetry-breaking field [32], as mentioned before. We expect that combining the latter with the method described in this paper would make an efficient algorithm for intermediate size error-corrected quantum computers.

## ACKNOWLEDGMENTS

We thank Matthias Troyer for their suggestions, and Anirban Narayan Chowdhury and Thomas Baker for stimulating

discussions. J.L. acknowledges support from the NSERC Canada Graduate Scholarships and the FRQNT programs of scholarships. Computing resources were provided by Compute Canada and Calcul Québec.

## APPENDIX A: SIMULATION METHOD

In this section, we provide details on the simulation methods used to solve the eigenproblem of the Hubbard models of different sizes. For small systems, where this is possible, we use exact diagonalization; otherwise we use DMRG to find an approximate solution. The goal of this section is to enable reproduction of the results.

### 1. Exact diagonalization

The exact results are computed with the Lanczos algorithm [42]. To reduce the matrix size, which scales as  $4^N$  where  $N$  is the number of sites, we only consider the subspace of interest where the numbers of up ( $N_\uparrow$ ) and down ( $N_\downarrow$ ) spins are fixed, leading to a matrix size

$$D = \binom{N!}{N_\uparrow!(N - N_\uparrow)!} \binom{N!}{N_\downarrow!(N - N_\downarrow)!}. \quad (\text{A1})$$

We chose the number of fermions to represent the worst-case scenario, i.e., the smallest gap. For the small lattices, we found that the smallest gap of the Hubbard model is at roughly 10% doping above (or equivalently below) half filling. We choose to keep this filling factor in the hope to be close to the smallest gap (at fixed number of spins) which correspond to hard problems in the adiabatic setting. For example, for a 64-site lattice, half filling means 32 up spins and 32 down spins. At 10% doping above (resp. below) half filling, we will have 35 or 36 (resp. 29 or 28) up and down spins.

### 2. Density-Matrix Renormalization Group

For bigger lattice sizes, exact diagonalization is intractable. We used DMRG, a tensor network method, to approximately solve the eigenproblem. See Ref. [43] for a review. The method has been implemented using the ITENSOR C++ library [45]. To preserve the number of up and down spins, we used the IQTensor object.

The  $m \times k$  lattices (for a total of  $km = N$  sites and  $m > k$ ) are mapped to a MPS (see Fig. 5).

The energy error goal (tolerated error on the energy) is set to  $10^{-9}$ , the number of sweep is set to 30 to reach the ground state and to 40 to reach the first-excited state. Note that to find the first-excited state, we need to set a penalty  $\tilde{\Delta} \in \mathbb{R}_+^*$  to the ground state by running the DMRG again for the following

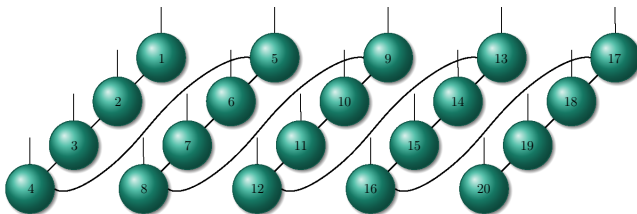


FIG. 5. MPS of the  $m \times k$  lattice ( $m = 5$ ,  $k = 4$ ).

TABLE I. The schedule of the sweep parameter given to the IQTensor DMRG function (maximal bound dimension—maxm, minimal bound dimension—minm, the truncation error cutoff—Cutoff, the number of Davidson iterations—niter and the noise term—Noise). The last row is the parameter used for the remaining sweeps.

maxm	minm	Cutoff	niter	Noise
2	1	$10^{-5}$	2	$10^{-1}$
2	1	$10^{-6}$	2	$10^{-1}$
4	1	$10^{-7}$	2	$10^{-2}$
4	1	$10^{-8}$	2	$10^{-2}$
8	1	$10^{-8}$	2	$10^{-3}$
8	1	$10^{-8}$	2	$10^{-3}$
16	1	$10^{-8}$	2	$10^{-4}$
16	1	$10^{-9}$	2	$10^{-4}$
32	1	$10^{-9}$	2	$10^{-5}$
32	1	$10^{-10}$	2	$10^{-5}$
64	1	$10^{-10}$	2	$10^{-6}$
64	1	$10^{-10}$	2	$10^{-6}$
64	1	$10^{-10}$	2	$10^{-6}$
128	1	$10^{-10}$	2	$10^{-7}$
128	1	$10^{-10}$	2	$10^{-7}$
256	1	$10^{-10}$	2	$10^{-7}$
256	1	$10^{-10}$	2	$10^{-8}$
512	1	$10^{-10}$	2	$10^{-8}$
512	1	$10^{-10}$	2	$10^{-8}$
1024	1	$10^{-10}$	2	$10^{-9}$

Hamiltonian:

$$H' = H + \tilde{\Delta} |\psi_0\rangle\langle\psi_0|, \quad (\text{A2})$$

where  $|\psi_0\rangle$  is the ground state of  $H$ . By a trial and error approach, we found that setting  $\tilde{\Delta}$  to the absolute value of the ground-state energy if that value is higher than 1, and to 10 if it is smaller than or equal to 1 give better result and convergence rate.

For the other ITENSOR parameters, we built a ramp to avoid getting stuck in a local minimum (see Table I).

## APPENDIX B: SCALING

The scaling of the quantum adiabatic algorithm is bounded above by  $O(\frac{1}{\Delta^2})$  [41], where  $\Delta$  is the minimal gap of the time-dependent Hamiltonian

$$H(t) = \left(1 - \frac{t}{\tau}\right)H_0 + \frac{t}{\tau}H_f. \quad (\text{B1})$$

$\tau$  is the adiabatic parameter,  $H_0$  is the initial Hamiltonian with an easy-to-prepare ground state, and  $H_f$  is the final Hamiltonian with the target ground state. In the discrete version, the equivalent of the delay schedule  $\tau$  is defined by the density of steps and the precision of the energy measurements.

One could use a faster schedule in some instances and obtain a better scaling than the theoretical bound. The procedure introduced in the main paper results in a faster schedule than constant-step linear interpolation. The minimal TTS finds a trade-off between a long high-probability computation and short low-probability computation repeated multiple times.

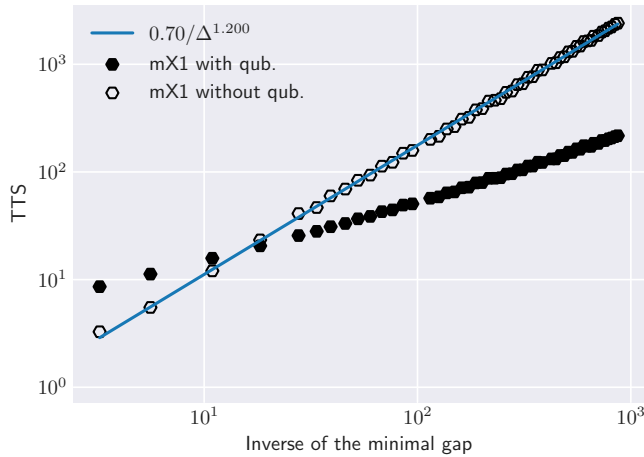


FIG. 6. TTS obtained with and without qubitization (empty and filled symbols reps.) on chain,  $m \times 1$ , lattices. The units of the TTS are set by the natural units of the Hubbard model ( $T_{\text{hub}} = 1$ ). The  $x$  axis corresponds to the minimal gap of the original problem also in the natural units of the Hubbard model (before qubitization).

We expect the scaling corresponding to the density of step to be at worst  $1/\Delta$  for all instances.

By using qubitization, we modified both the gap and states used for the evolution. The change in the gap has an obvious impact on the scaling, changing a factor from  $1/\Delta$  to  $1/\cos(1 - \bar{\Delta})$ , where  $\bar{\Delta}$  is the gap of the normalized Hamiltonian. Qubitization will not change the density of steps used but will affect the success probability of each step since the states used are the eigenstates of  $W_j = \exp\{i \cos(\bar{H}_j)\}$ .

An exact study of the scaling including all contributions (qubitization, schedule choice, rewind protocol) is hard to perform. To get a sense of the scaling of our simulation, see Figs. 6 and 7. Because the simulation results appear to de-

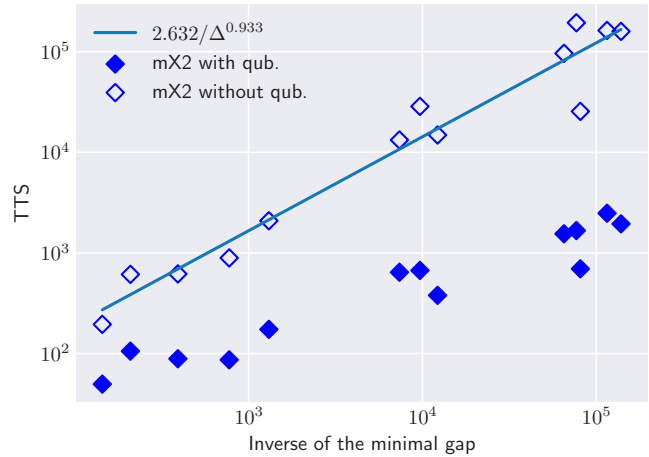


FIG. 7. TTS obtained with and without qubitization (empty and filled symbols reps.) on ladder,  $m \times 2$ , lattices. The units of the TTS are set by the natural units of the Hubbard model ( $T_{\text{hub}} = 1$ ). The  $x$  axis corresponds to the minimal gap of the original problem also in the natural units of the Hubbard model (before qubitization).

pend on lattice depth, we focused on ladder and chain lattices ( $m \times 2$  and  $m \times 1$ ). The horizontal axis is the inverse minimal gap of the initial problem, thus, before qubitization. Since the qubitization increases the gap size, it also reduces greatly the cost of the adiabatic evolution (filled symbols). The scaling is better than  $1/\Delta^2$  in all cases we studied. The density of steps should be bounded by  $1/\Delta$ . However, the different trends from the different lattice shapes indicate that the gap is not the only determinant factor in our scaling. The implicit term in the TTS is nontrivial. From the numerics, there is a significant speedup, but we cannot distinguish what is general from what is specific to the system used. Thus, the fitting curves on the figures should be considered as guides, not as an actual scaling of the method.

- 
- [1] R. P. Feynman, *Int. J. Theor. Phys.* **21**, 467 (1982).  
 [2] S. Lloyd, *Science* **273**, 1073 (1996).  
 [3] D. Aharonov and A. Ta-Shma, in *Proceedings of the 35th Annual ACM Symposium on Theory Computation* (ACM, New York, 2003).  
 [4] A. M. Childs and Y. Su, *Phys. Rev. Lett.* **123**, 050503 (2019).  
 [5] D. W. B. Ryan Babbush, Y. R. Sanders, I. D. Kivlichan, A. Scherer, A. Y. Wei, P. J. Love, and A. Aspuru-Guzik, *Quantum Sci. Technol.* **3**, 015006 (2018).  
 [6] Y. Cao, J. Romero, J. P. Olson, M. Degroote, P. D. Johnson, M. Kieferová, I. D. Kivlichan, T. Menke, B. Peropadre, N. P. Sawaya, S. Sim, L. Veis, and A. Aspuru-Guzik, *Chem. Rev.* **119**, 10856 (2019).  
 [7] D. Poulin, M. B. Hastings, D. Wecker, N. Wiebe, A. C. Doherty, and M. Troyer, *Quantum Inf. Comput.* **15**, 0361 (2015).  
 [8] D. Wecker, B. Bauer, B. K. Clark, M. B. Hastings, and M. Troyer, *Phys. Rev. A* **90**, 022305 (2014).  
 [9] S. P. Jordan, K. S. M. Lee, and J. Preskill, *Science* **336**, 1130 (2012).  
 [10] S. Raesi, N. Wiebe, and B. C. Sanders, *New J. Phys.* **14**, 103017 (2012).  
 [11] D. Berry and A. Childs, *Quantum Inf. Comput.* **12**, 29 (2012).  
 [12] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, *Phys. Rev. Lett.* **114**, 090502 (2015).  
 [13] R. Babbush, D. W. Berry, I. D. Kivlichan, A. Y. Wei, P. J. Love, and A. Aspuru-Guzik, *New J. Phys.* **18**, 033032 (2016).  
 [14] E. Campbell, *Phys. Rev. Lett.* **123**, 070503 (2019).  
 [15] G. H. Low and I. L. Chuang, *Phys. Rev. Lett.* **118**, 010501 (2017).  
 [16] A. Y. Kitaev, A. H. Shen, and M. N. Vyalıy, *Classical and Quantum Computation*, Graduate Studies in Mathematics (American Mathematical Society, Providence, 2002).  
 [17] J. Kempe, A. Kitaev, and O. Regev, *SIAM J. Comput.* **35**, 1070 (2006).  
 [18] J. Kempe and O. Regev, *Quantum Inf. Comput.* **3**, 258 (2003).  
 [19] R. Oliveira and B. M. Terhal, *Quantum Inf. Comput.* **8**, 0900 (2008).  
 [20] S. Boixo, E. Knill, and R. Somma, *Quantum Inf. Comput.* **9**, 833 (2009).  
 [21] K. Temme, T. Osborne, K. Vollbrecht, D. Poulin, and F. Verstraete, *Nature (London)* **471**, 87 (2011).

- [22] M.-H. Yung and A. Aspuru-Guzik, *Proc. Natl. Acad. Sci. USA* **109**, 754 (2012).
- [23] I. D. Kivlichan, C. Gidney, D. W. Berry, N. Wiebe, J. McClean, W. Sun, Z. Jiang, N. Rubin, A. Fowler, A. Aspuru-Guzik, R. Babbush, and H. Neven, *Quantum* **4**, 296 (2020).
- [24] D. Poulin and P. Wocjan, *Phys. Rev. Lett.* **102**, 130503 (2009).
- [25] J. Whitfield, J. Biamonte, and A. Aspuru-Guzik, *Mol. Phys.* **109**, 735 (2010).
- [26] R. Jones, *Rev. Mod. Phys.* **87**, 897 (2015).
- [27] P. H. Acioli, *J. Mol. Struct.: THEOCHEM* **394**, 75 (1997).
- [28] R. Orús, *Ann. Phys. (NY)* **349**, 117 (2014).
- [29] E. Farhi, J. Goldstone, S. Gutmann, J. Lapan, A. Lundgren, and D. Preda, *Science* **292**, 472 (2001).
- [30] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, [arXiv:quant-ph/0001106](https://arxiv.org/abs/quant-ph/0001106) (2000).
- [31] D. W. Berry, M. Kieferová, A. Scherer, Y. R. Sanders, G. H. Low, N. Wiebe, C. Gidney, and R. Babbush, *npj Quantum Inf.* **4**, 22 (2018).
- [32] D. Wecker, M. B. Hastings, N. Wiebe, B. K. Clark, C. Nayak, and M. Troyer, *Phys. Rev. A* **92**, 062318 (2015).
- [33] E. Farhi, J. Goldstone, and S. Gutmann, [arXiv:1411.4028](https://arxiv.org/abs/1411.4028).
- [34] A. Kitaev, Quantum measurements and the Abelian stabilizer problem, [arXiv:quant-ph/9511026](https://arxiv.org/abs/quant-ph/9511026).
- [35] R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca, *Proc. R. Soc. London, Ser. A* **454**, 339 (1998).
- [36] D. S. Abrams and S. Lloyd, *Phys. Rev. Lett.* **83**, 5162 (1999).
- [37] D. Poulin, A. Kitaev, D. S. Steiger, M. B. Hastings, and M. Troyer, *Phys. Rev. Lett.* **121**, 010501 (2018).
- [38] T. F. Rønnow, Z. Wang, J. Job, S. Boixo, S. V. Isakov, D. Wecker, J. M. Martinis, D. A. Lidar, and M. Troyer, *Science* **345**, 420 (2014).
- [39] C. Marriott and J. Watrous, *Comput. Complex.* **14**, 122 (2005).
- [40] G. H. Low and I. L. Chuang, *Quantum* **3**, 163 (2019).
- [41] W. van Dam, M. Mosca, and U. Vazirani, in *Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science, Newport Beach, CA, USA* (IEEE, Piscataway, NJ, 2001), pp. 279–287.
- [42] C. Lanczos, *J. Res. Natl. Bur. Stand.* **45**, 255 (1950).
- [43] U. Schollwöck, *Ann. Phys. (NY)* **326**, 96 (2011).
- [44] E. Stoudenmire and S. R. White, *Annu. Rev. Condens. Matter Phys.* **3**, 111 (2012).
- [45] ITensor Library (version 2.0.11), <http://itensor.org>.
- [46] A. M. Childs, D. Maslov, Y. Nam, N. J. Ross, and Y. Su, *Proc. Natl. Acad. Sci. USA* **115**, 9456 (2018).
- [47] J. Lemieux, B. Heim, D. Poulin, K. Svore, and M. Troyer, *Quantum* **4**, 287 (2020).

## Chapitre 3

# Algorithme de marche aléatoire

*The designation "random walk" seems apt since a realization of the process describes the path of a person (suitably intoxicated) moving randomly one step forward or backward*

— Karlin, 2014 [36]

Les méthodes d'échantillonnage telles que celle de Monte-Carlo sont fréquemment utilisées pour résoudre une variété de problèmes. On pense notamment à l'intégration numérique et l'optimisation. Les chaînes de Markov, par exemple, permettent l'échantillonnage d'une distribution de probabilité difficile à calculer. Étant donné la nature de l'ordinateur quantique, l'une des applications envisagées est l'échantillonnage d'une distribution de probabilité représentée par un état.

En 2004, Szegedy a introduit une discrétisation des marches bipartites réversibles [37]. Toutefois, cette procédure ne permet pas de préparer directement la distribution de probabilité visée. On se doit de combiner un tel processus à une méthode comme le recuit simulé (approche similaire à l'évolution adiabatique) afin d'obtenir un état représentant notre distribution. De plus, l'algorithme de Szegedy fait usage d'un oracle. Nous nous sommes donc intéressés entre autres à optimiser, voire reformuler, l'opérateur de marche de manière à avoir une implémentation explicite et efficace.

L'algorithme de marche aléatoire élaboré pour cette thèse est basé sur plusieurs concepts clefs, dont la quantification de marches aléatoires décrite par Szegedy

(pour une rétrospective, voir [A.4](#)), l'algorithme classique de Metropolis-Hastings [[38](#)] (voir [A.3](#)) et une discrétisation de l'algorithme d'évolution adiabatique [[27](#)].

### 3.1 Article

Pour ce projet, j'ai fait les simulations et les calculs afin d'obtenir les résultats des figures 1 et 2, ainsi que des tableaux 1 et 2. J'ai développé la décomposition en circuit de l'opérateur de marche –incluant l'une des variantes par rapport à l'opérateur de Szegedy. J'ai également démontré la correspondance de cette nouvelle formulation avec l'opérateur de Szegedy et, de manière plus générale, avec les marches classiques et quantiques. J'ai aussi développé l'usage heuristique de l'algorithme. Finalement, j'ai aussi participé à la révision de l'article.

Cet article a été publié au journal *Quantum* :  
Jessica Lemieux, Bettina Heim, David Poulin, Krysta Svore et Matthias Troyer.  
Efficient Quantum Walk Circuits for Metropolis-Hastings Algorithm. *Quantum*, 4,  
287 (2020), doi :[10.22331/q-2020-06-29-287](https://doi.org/10.22331/q-2020-06-29-287).

# Efficient Quantum Walk Circuits for Metropolis-Hastings Algorithm

Jessica Lemieux<sup>1</sup>, Bettina Heim<sup>2</sup>, David Poulin<sup>1,3</sup>, Krysta Svore<sup>2</sup>, and Matthias Troyer<sup>2</sup>

<sup>1</sup>Département de Physique & Institut Quantique, Université de Sherbrooke, Québec, Canada

<sup>2</sup>Quantum Architecture and Computation Group, Microsoft Research, Redmond, WA 98052, USA

<sup>3</sup>Canadian Institute for Advanced Research, Toronto, Ontario, Canada M5G 1Z8

**We present a detailed circuit implementation of Szegedy’s quantization of the Metropolis-Hastings walk. This quantum walk is usually defined with respect to an oracle. We find that a direct implementation of this oracle requires costly arithmetic operations. We thus reformulate the quantum walk, circumventing its implementation altogether by closely following the classical Metropolis-Hastings walk. We also present heuristic quantum algorithms that use the quantum walk in the context of discrete optimization problems and numerically study their performances. Our numerical results indicate polynomial quantum speedups in heuristic settings.**

Markov chain Monte Carlo (MCMC) methods are a cornerstone of modern computation, with applications ranging from computational science to machine learning. The key idea is to sample a distribution  $\pi_x$  by constructing a random walk  $\mathcal{W}$  which reaches this distribution at equilibrium  $\mathcal{W}\pi = \pi$ . One important characteristic of a Markov chain is its mixing time, the time it requires to reach equilibrium. This mixing time is governed by the inverse spectral gap of  $\mathcal{W}$ , where the spectral gap  $\Delta$  is defined as the difference between its two largest eigenvalues. The runtime of a MCMC algorithm is thus determined by the product of the mixing time and the time required to implement a single step of the walk.

Szegedy [28] presented a general method to quantize reversible walks, resulting in a unitary transformation  $U_{\mathcal{W}}$ . The eigenvalues  $e^{i\theta_j}$  of a unitary matrix all lie on the unit complex circle, and we choose  $0 = \theta_0 \leq \theta_1 \leq \theta_2 \leq \dots$ . The steady state  $|\pi\rangle$  of the quantum walk is essentially a coherent version  $|\pi\rangle = \sum_x \sqrt{\pi_x}|x\rangle$  of the classical equilibrium distribution  $\pi$ . The main feature of the quantum walk is that its spectral

gap  $\delta := \theta_1 \geq \sqrt{\Delta}$  is quadratically larger than its classical counterpart. Combined with the quantum adiabatic algorithm [1, 6, 9], this yields a quantum algorithm to reach the steady state that scales quadratically faster with  $\Delta$  than the classical MCMC algorithm [27].

While at first glance this is an important advantage with far-reaching applications, additional considerations must be taken into account to determine if quantum walks offer a significant speedup for any specific application. One of the reasons is that it could take significantly longer to implement a single step  $U_{\mathcal{W}}$  of the quantum walk than to implement a step  $\mathcal{W}$  of the classical walk. Thus, quantum walks are more likely to offer advantages in situations with extremely long equilibration times. Moreover, we must address the fact that classical walks are often used heuristically out of equilibrium. When training a neural network for instance, where a MCMC method called stochastic gradient descent is used to minimize a cost function, it is in practice often not necessary to reach the true minimum, and thus the MCMC runs in time less than its mixing time. Similarly, simulated annealing is typically used heuristically with cooling schedules far faster than prescribed by provable bounds – and combined with repeated restarts. Such heuristic applications further motivate the constructions of efficient implementations of  $U_{\mathcal{W}}$ , and the development of heuristic methods for quantum computers.

This article addresses these two points. First, we present a detailed realization and cost analysis of the quantum walk operator for the special case of a Metropolis-Hastings walk [12, 23]. This is a widespread reversible walk, whose implementation only requires knowledge of the relative populations  $\pi_x/\pi_y$  of the equilibrium distribution. While Szegedy’s formulation of the quan-

arXiv:1910.01659v2 [quant-ph] 25 Jun 2020



tum walk builds on a classical walk oracle, our implementation circumvents its direct implementation, which would require costly arithmetic operations. Instead, we directly construct a related but different quantum unitary walk operator with an effort to minimize circuit depth. Second, we suggest heuristic uses of this oracle inspired by the adiabatic algorithm, and study their performances numerically.

## 1 Preliminaries

### 1.1 Quantum Walk

We define a classical walk on a  $d$ -dimensional state space  $\mathcal{X} = \{x\}$  by a  $d \times d$  transition matrix  $\mathcal{W}$  where the transition probability  $x \rightarrow y$  is given by matrix element  $\mathcal{W}_{yx}$ . Thus, the walk maps the distribution  $p$  to the distribution  $p' = \mathcal{W}p$ , where  $p'_y = \sum_x \mathcal{W}_{yx} p_x$ . An aperiodic walk is *irreducible* if every state in  $\mathcal{X}$  is accessible from every other state in  $\mathcal{X}$ , which implies the existence of a unique equilibrium distribution  $\pi = \mathcal{W}\pi$ . Finally, a walk is *reversible* if it obeys the detailed balance condition

$$\mathcal{W}_{yx}\pi_x = \mathcal{W}_{xy}\pi_y. \quad (1)$$

We now explain how to quantize a reversible classical walk  $\mathcal{W}$ .

Szegedy's quantum walk [28] is formulated in an oracle setting. For a classical walk  $\mathcal{W}$ , it assumes a unitary transformation  $W$  acting on a Hilbert space  $\mathbb{C}^d \otimes \mathbb{C}^d$  with the following action

$$W|x\rangle \otimes |0\rangle = |w_x\rangle \otimes |x\rangle =: |\phi_x\rangle, \quad (2)$$

where  $|w_x\rangle := \sum_y \sqrt{\mathcal{W}_{yx}}|y\rangle$ . Define  $\Pi_0$  as the projector onto the subspace  $\mathcal{E}_0$  spanned by states  $\{|x\rangle \otimes |0\rangle\}_{x=1}^d$ . Combining  $W$  to the reflection  $R = 2\Pi_0 - I$  and the swap operator  $\Lambda$ , we can construct the quantum walk defined by

$$U_{\mathcal{W}} := RW^\dagger \Lambda W \quad (3)$$

$$= (2\Pi_0 - 1)W^\dagger \Lambda W. \quad (4)$$

Szegedy's walk is defined as  $\Lambda W(RW^\dagger \Lambda W)RW^\dagger$ , so it is essentially the square of the operator  $U_{\mathcal{W}}$  we have defined, but this will have no consequence on what follows aside from a minor simplification.

To analyze the quantum walk  $U_{\mathcal{W}}$ , let us define the state  $|\psi_x\rangle := \Lambda|\phi_x\rangle = |x\rangle \otimes |w_x\rangle$  and consider

the operator

$$X := \Pi_0 W^\dagger \Lambda W \Pi_0 \quad (5)$$

$$= \sum_{xy} \langle \phi_y | \psi_x \rangle |y\rangle \langle x| \otimes |0\rangle \langle 0| \quad (6)$$

$$= \sum_{xy} \sqrt{\mathcal{W}_{xy}\mathcal{W}_{yx}} |y\rangle \langle x| \otimes |0\rangle \langle 0|. \quad (7)$$

At this point, in order to use detailed balance condition of Eq. (1), we need to assume that the walk is reversible to obtain

$$X = \sum_{xy} \sqrt{\frac{\pi_x}{\pi_y}} \mathcal{W}_{yx} |y\rangle \langle x| \otimes |0\rangle \langle 0|, \quad (8)$$

or, if we restrict the operator  $X$  to its support  $\mathcal{E}_0$ , we get in matrix notation  $X = \text{diag}(\pi^{-\frac{1}{2}})\mathcal{W}\text{diag}(\pi^{\frac{1}{2}})$ . The matrices  $X$  and  $\mathcal{W}$  are thus similar so they have the same eigenvalues. Define its eigenvectors

$$X|\tilde{\gamma}_k\rangle = \lambda_k|\tilde{\gamma}_k\rangle, \quad (9)$$

where  $\lambda_k$  are the eigenvalues of  $\mathcal{W}$ . Because the operator  $X$  is obtained by projecting the operator  $W^\dagger \Lambda W$  onto the subspace  $\mathcal{E}_0$ , its eigenvectors with non-zero eigenvalues in the full Hilbert space must have the form  $|\gamma_k\rangle = |\tilde{\gamma}_k\rangle \otimes |0\rangle$ .

If we consider the action of  $W^\dagger \Lambda W$  without those projections, we get

$$W^\dagger \Lambda W |\gamma_k\rangle = \lambda_k |\gamma_k\rangle - \beta_k |\gamma_k^\perp\rangle \quad (10)$$

where  $|\gamma_k^\perp\rangle$  is orthogonal to the subspace  $\mathcal{E}_0$ , so in particular it is orthogonal to all the vectors  $|\gamma_{k'}\rangle$ . Finally, because  $W^\dagger \Lambda W$  is a unitary, we also obtain that the  $|\gamma_k^\perp\rangle$  are orthogonal to each other and that  $\beta_k = \sqrt{1 - |\lambda_k|^2}$ . This implies that the vectors  $\{|\gamma_k\rangle, |\gamma_k^\perp\rangle\}$  are all mutually orthogonal and that  $W^\dagger \Lambda W$  is block diagonal in that basis.

Given the above observations, it is straightforward to verify that

$$U_{\mathcal{W}} |\gamma_k\rangle = \lambda_k |\gamma_k\rangle + \sqrt{1 - |\lambda_k|^2} |\gamma_k^\perp\rangle \quad (11)$$

$$U_{\mathcal{W}} |\gamma_k^\perp\rangle = \sqrt{1 - |\lambda_k|^2} |\gamma_k\rangle - \lambda_k |\gamma_k^\perp\rangle, \quad (12)$$

so the eigenvalues of  $U_k$  on the subspace spanned by  $\{|\gamma_k\rangle, |\gamma_k^\perp\rangle\}$  are  $e^{\pm i\theta_k}$  where  $\cos \theta_k = \lambda_k$  with corresponding eigenvectors  $|\gamma_k^\pm\rangle = \frac{1}{\sqrt{2}}(|\gamma_k\rangle \pm i|\gamma_k^\perp\rangle)$ .

## 1.2 Adiabatic state preparation

We can use quantum phase estimation [16] to measure the eigenvalues of  $U_{\mathcal{W}}$ . In particular, we want this measurement to be sufficiently accurate to resolve the eigenvalue  $\theta = 0$ , or equivalently  $\lambda_k = 1$ , from the rest of the spectrum. Assuming that the initial state is supported on the subspace  $\mathcal{E}_0$ , the spectral gap of  $U_{\mathcal{W}}$  is  $\delta = \theta_1 = \arccos(\lambda_1) = \arccos(1 - \Delta) \sim \sqrt{\Delta}$ , so we only need about  $1/\sqrt{\Delta}$  applications of  $U_{\mathcal{W}}$  to realize that measurement. This is quadratically faster than the classical mixing time  $1/\Delta$ , which is the origin of the quadratic quantum speed-up.

A measurement outcome corresponding to  $\theta = 0$  would produce the coherent stationary distribution  $|\pi\rangle \otimes |0\rangle := \sum_x \sqrt{\pi_x} |x\rangle \otimes |0\rangle$ . Indeed, first note that for any  $|\psi\rangle$  such that  $X(|\psi\rangle \otimes |0\rangle) = |\psi\rangle \otimes |0\rangle$ , Eq. (10) implies that  $U_{\mathcal{W}}(|\psi\rangle \otimes |0\rangle) = |\psi\rangle \otimes |0\rangle$ . We can verify that this condition holds for  $|\psi\rangle = |\pi\rangle$ :

$$X \sum_x \sqrt{\pi_x} |x\rangle \otimes |0\rangle = \sum_{xy} \sqrt{\frac{\pi_x}{\pi_y}} \mathcal{W}_{yx} \sqrt{\pi_x} |y\rangle \otimes |0\rangle \quad (13)$$

$$= \sum_{xy} \mathcal{W}_{xy} \sqrt{\pi_y} |y\rangle \otimes |0\rangle \quad (14)$$

$$= \sum_y \sqrt{\pi_y} |y\rangle \otimes |0\rangle \quad (15)$$

where we have used detailed balance Eq. (1) in the second step and  $\sum_x \mathcal{W}_{xy} = 1$  in the last step.

From an initial state  $|\psi\rangle \otimes |0\rangle = \sum_k \alpha_k |\gamma_k\rangle$ , the probability of that measurement outcome is  $|\langle \psi | \pi \rangle|^2 = |\alpha_0|^2$ . Therefore, the initial state  $|\psi\rangle$  must be chosen with a large overlap with the fixed point to ensure that this measurement outcome has a non-negligible chance of success. If no such state can be efficiently prepared, one can use adiabatic state preparation [1, 9] to increase the success probability. In its discrete formulation [27] inspired by the quantum Zeno effect, we can choose a sequence of random walks  $\mathcal{W}^0, \mathcal{W}^1, \dots, \mathcal{W}^L = \mathcal{W}$  with coherent stationary distributions  $|\pi^j\rangle$ . The walks are chosen such that  $|\pi^0\rangle$  is easy to prepare and consecutive walks are nearly identical, so that  $|\langle \pi^j | \pi^{j+1} \rangle|^2 \geq 1 - \frac{1}{L}$  [27]. Thus, the sequence of  $L$  measurements of the eigenstate of the corresponding quantum walk operators  $U_{\mathcal{W}^j}$  all yield the outcomes  $\theta = 0$  with probability  $(1 - \frac{1}{L})^L \sim \frac{1}{e}$ , which results in the desired state. The overall complexity of this al-

gorithm is

$$C \sum_{j=1}^L \frac{1}{\delta_j} \quad (16)$$

where  $\delta_j$  is the spectral gap of the  $j$ -th quantized walk  $\mathcal{W}^j$  and  $C$  is the time required to implement a single quantum walk operator.

## 1.3 Metropolis-Hastings Algorithm

The Metropolis-Hastings algorithm [12, 23] uses a special class of Markov chains which obey detailed balance Eq. (1) by construction. The basic idea is to break the calculation of the transition probability  $x \rightarrow y$  in two steps. First, a transition from  $x$  to  $y \neq x$  is proposed with probability  $T_{yx}$ . Then, this transition is accepted with probability  $A_{yx}$  and otherwise rejected, in which case, the state remains  $x$ . The overall transition probability is thus

$$\mathcal{W}_{yx} = \begin{cases} T_{yx} A_{yx} & \text{if } y \neq x \\ 1 - \sum_y T_{yx} A_{yx} & \text{if } y = x. \end{cases} \quad (17)$$

The detailed balance condition Eq. (1) becomes

$$R_{xy} := \frac{A_{yx}}{A_{xy}} = \frac{\pi_y T_{xy}}{\pi_x T_{yx}}, \quad (18)$$

which in the Metropolis-Hastings algorithm is solved with the choice

$$A_{yx} = \min(1, R_{xy}). \quad (19)$$

We note that our quantum algorithm can also be applied to the Glauber, or heat-bath, choice [10, 30]

$$A_{yx} = \frac{1}{1 + R_{yx}}. \quad (20)$$

The Metropolis-Hastings algorithm is widely used to generate a Boltzmann distribution with applications in statistical physics and machine learning. Given a real energy function  $E(x)$  on the configuration space  $X$ , the Boltzmann distribution at inverse temperature  $\beta$  is defined as  $\pi_x^\beta = \frac{1}{Z(\beta)} e^{-\beta E(x)}$  where the partition function  $Z(\beta)$  ensures normalization. In this setting, it is common practice to choose a symmetric proposed transition probability  $T_{yx} = T_{xy}$ , so the acceptance probability depends only on the energy difference

$$A_{yx} = \min\left(1, e^{\beta[E(x) - E(y)]}\right). \quad (21)$$

Note that the Metropolis-Hastings algorithm can be applied to quantum mechanical Hamiltonians [29], where it can also benefit from a quadratic speed-up using Szegedy's quantization procedure [31].

## 2 Circuit for Walk operator

Quantum algorithms built from quantization of classical walks [2, 21, 27, 28] usually assume an oracle formulation of the walk operator, where the ability to implement the transformation  $W$  of Eq. (2) is taken for granted. As we discuss below in Appendix A, this transformation requires costly arithmetic operations. One of the key innovations of this article is to provide a detailed and simplified implementation of a walk operator along with a detailed cost analysis of Metropolis-Hastings walks. As it will become apparent, our implementation circumvents the use of  $W$  altogether.

For concreteness, we will assume a  $(k, d)$ -local Ising model, where  $X = \{+1, -1\}^n$ , and the energy function takes the simple form

$$E(x) = \sum_{\ell} J_{\ell} \prod_{s \in \Omega_{\ell}} x_s, \quad (22)$$

where  $\Omega_{\ell}$  are subsets of at most  $k$  Ising spins,  $J_{\ell}$  are real coupling constants where  $\ell$  ranges over all the possible couplings (from 1 to  $\frac{nd}{k}$ ), and each spin interacts with at most  $d$  other spins. Note that for  $k = 2$  and  $d \geq 3$ , finding the ground state is an NP-hard problem[3].

As it is always the case for Ising models, we will assume that the proposed transitions of the Metropolis-Hastings walk are obtained by choosing a random set of spins and inverting their signs. In other words,  $T_{yx} = f(x \cdot y)$  where the product is taken bit by bit and where  $f(z)$  is some simple probability distribution on  $X - \{1^n\}$  (it does not contain a trivial move), so  $T_{yx}$  is clearly symmetric. The distribution  $f(z)$  is sparse, in the sense that it has only  $N \in O(n)$  non-zero entries.

For concreteness, we will suppose that  $f$  is uniform over some set  $\mathcal{M}$  of moves, with  $|\mathcal{M}| = N$ :

$$T_{xy} = \begin{cases} \frac{1}{N} & \text{if } z = x \cdot y \in \mathcal{M} \\ 0 & \text{otherwise} \end{cases}. \quad (23)$$

The most common example consists of single-spin moves, where a single spin is chosen uniformly at random to be flipped. More generally, we

will suppose that moves are sparse in the sense that each move  $z_j \in \mathcal{M}$  flips a constant-bounded number of spins and that each spin belongs to a constant-bounded number of different moves. For  $j = 1, 2, \dots, N$ , we use  $f(j)$  as a shorthand for  $f(z_j)$ . With a further abuse of notation, we view  $z_j \in \mathcal{M}$  both as Ising spin configurations and as subsets of  $[n]$ , where the correspondence is given by the locations of  $-1$  spins in  $z_j$ .

A direct implementation of the unitary  $W$  generally requires costly quantum circuits involving arithmetic operations. The complexity arises from the need to uncompute a move register and a Boltzmann coin when implementing  $W$ . This turns out to be non-trivial and costly if a move is rejected. Consequently, we do not implement  $W$ , but instead present a circuit which is isometric to the entire walk operator  $U_{\mathcal{W}}$ , thus avoiding the problem. In other words, we construct a circuit for  $\tilde{U}_{\mathcal{W}} := Y^{\dagger} U_{\mathcal{W}} Y$  where  $Y$  maps

$$Y : |x\rangle \otimes |y\rangle \rightarrow \begin{cases} |x\rangle \otimes |x \cdot y\rangle & \text{if } x \cdot y \in \mathcal{M} \\ 0 & \text{otherwise.} \end{cases} \quad (24)$$

To minimize circuit depth, the second register above is encoded in a unary representation, so it contains  $N$  qubits and  $|z\rangle$  is encoded as  $|00\dots 0100\dots\rangle$  with a 1 at the  $z$ -th position. Since the state is already encoded in  $N$  qubits, unary encoding adds only a small multiplicative number of qubits compared to binary encoding. In addition to these two registers, the circuit acts on an additional coin qubit. Thus, we will denote the System, Move, and Coin registers with corresponding subscripts  $|x\rangle_S |z\rangle_M |b\rangle_C$ , and they contain  $n$ ,  $N$ , and 1 qubits respectively.

Our implementation of the walk operator combines four components:

$$\tilde{U}_{\mathcal{W}} = RV^{\dagger} B^{\dagger} F B V \quad (25)$$

where

$$V : |0\rangle_M \rightarrow \sum_j \sqrt{f(j)} |j\rangle_M, \quad (26)$$

$$B : |x\rangle_S |j\rangle_M |0\rangle_C \rightarrow |x\rangle_S |j\rangle_M \left( \sqrt{1 - A_{x \cdot z_j, x}} |0\rangle + \sqrt{A_{x \cdot z_j, x}} |1\rangle \right)_C, \quad (27)$$

$$F : |x\rangle_S |j\rangle_M |b\rangle_C \rightarrow |x \cdot z_j^b\rangle_S |j\rangle_M |b\rangle_C, \quad \text{and} \quad (28)$$

$$R : |0\rangle_M |0\rangle_C \rightarrow -|0\rangle_M |0\rangle_C, \\ |j\rangle_M |b\rangle_C \rightarrow |j\rangle_M |b\rangle_C \text{ for } (j, b) \neq (0, 0) \quad (29)$$

While these definitions differ slightly from the ones of Sec. 1.1, it can be verified straightforwardly that these realize the desired walk operator, similar to our discussion in Sec. 1.1. In what follows, we provide a complete description of each of these components, and their complexity is summarized in Table 1.

## 2.1 Move preparation $V$

Recall that the Move register is encoded in unary. For a general distribution  $f$ , the method of [26] can be adapted to realize the transformation Eq. (26). Here, we focus on the case of a uniform distribution.

To begin, suppose that  $N$  is a power of 2. Starting in the state  $|000\dots 01\rangle_M$ , the state  $\frac{1}{N}\sum_j |j\rangle_M$  (in unary) is obtained by applying a sequence of  $N$  gates  $\sqrt{\text{SWAP}}$  in a binary-tree fashion. To see this, recall that  $\sqrt{\text{SWAP}}|10\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$ .

The gate  $\sqrt{\text{SWAP}}$  is in the third level of the Clifford hierarchy, so it can be implemented exactly using a constant number of  $T$  gates. This represents a substantial savings compared to the method of [26] for a general distributions which requires arbitrary rotations obtained from costly gate synthesis.

When  $N$  is not a power of 2, in order to avoid costly rotations, we choose to pad the distribution with additional states and prepare a distribution  $\frac{1}{2^\ell}\sum_j^{2^\ell} |j\rangle_M$  where  $\ell = \lceil \log_2 N \rceil$ . The states  $j = 1, 2, \dots, N$  encode the  $N$  moves  $\mathcal{M}$  of the classical walk  $x \rightarrow y = x \cdot z_j$ , while the additional states  $j > N$  correspond to trivial moves  $x \rightarrow x$ . This padding has the effect of slowing down the classical walk by a factor  $2^\ell/N < 2$ , and hence the quantum walk by a factor less than  $\sqrt{2}$ , which is less than the additional cost of preparing a uniform distribution over a range which is not a power of 2.

## 2.2 Spin flip $F$

The operator  $F$  of Eq. (28) flips a set of system spins  $z_j$  conditioned on the coin qubit and on the  $j$ -th qubit of the move register being in state 1. This can be implemented with at most  $Nc$  Toffoli gates (controlled-controlled-NOT), where the constant  $c$  upper-bounds the number of spins that are flipped by a single move of  $\mathcal{M}$ . The coin register acts as one control for each gate, the  $j$ -th bit of the move register acts as the other control,

and the targets are the system register qubits that are in  $z_j$ , for  $j = 1, 2, \dots, N$ . No gate is applied to the padding qubits  $j > N$ .

This implementation has the disadvantage of being purely sequential. An alternative implementation uses  $\mathcal{O}(N)$  additional scratchpad qubits but is entirely parallel. The details of the implementation depends on the sparsity of the moves  $\mathcal{M}$ , and in general there is a tradeoff between the scratchpad size and the circuit depth. When the moves consist of single-spin flips for instance, this uses  $N$  CNOTs in a binary-tree fashion (depth  $\log_2 N$ ) to make  $N$  copies of the coin qubit. The Toffoli gates can then be applied in parallel for each move, and lastly the CNOTs are undone.

## 2.3 Reflection $R$

The transformation  $R$  of Eq. (29) is a reflection about the state  $|00\dots 0\rangle_M|0\rangle_C$ . Using standard phase kickback methods, it can be implemented with a single additional qubit in state  $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$  and an open-control $^{(N+1)}$ -NOT gate. The latter can be realized from  $4(N-1)$  serial Toffoli gates [4] and linear depth.

Since our goal is to minimize circuit depth, we use a different circuit layout that uses at most  $N$  ancillary qubits and  $4N$  Toffoli gates to realize the  $(N+1)$ -fold controlled-not. The circuit once again proceeds in a binary tree fashion, dividing the set of  $N+1$  qubits into  $(N+1)/2$  pairs and applying a Toffoli gate between every pair with a fresh ancilla in state 0 as the target. The ancillary qubit associated to a given pair is in state 0 if and only if both qubits of the pair are in state 0. The procedure is repeated for the  $(N+1)/2$  ancillary qubits, until a single bit indicates if all qubits are in state 0. The ancillary bits are then uncomputed. Thus, the total depth in terms of gates in 3rd level of the Clifford hierarchy is  $2\log_2 N$ .

## 2.4 Boltzmann coin $B$

The Boltzmann coin given in Eq. (27) is the most expensive component of the algorithm, simply because it is the only component which requires rotations by arbitrary angles. Specifically, conditioned on move qubit  $j$  being 1 and the system register being in state  $x$ , the coin register under-

Gate	3L depth	3L count	Total depth	Qubits
$V$	$\log_2 N + 1$	$2N$	$\log_2 N + 1$	$2N$
$F$	1	$N$	$\log_2 N + 1$	$2N + n$
$R$	$2 \log N$	$4N$	$2 \log N$	$2N$
$B$	$\mathcal{O}(2^d \log \frac{1}{\epsilon})$	$\mathcal{O}(N2^d \log \frac{1}{\epsilon})$	$\mathcal{O}(\log N2^d \log \frac{1}{\epsilon})$	$2N + n + 2$

Table 1: Upper bound on the complexity of each component of the walk operator. The cost is measured in terms of number of gates in the 3rd level of the Clifford hierarchy, which is equivalent to  $T$  depth up to a small multiplicative factor. These are evaluated for a  $(k, d)$ -local Ising model with moves consisting of single-spin flips, in which case  $N = n$ . These costs could otherwise increase by a constant multiplicative amount determined by  $k, d$  and the sparsity of the moves  $z \in \mathcal{M}$ .

goes a rotation by an angle

$$\theta_{x,j} = \arcsin \left( \sqrt{\min\{e^{-\beta\Delta_j(x)}, 1\}} \right) \quad (30)$$

for Metropolis-Hastings or

$$\theta_{x,j} = \arcsin \left( \frac{1}{1 + e^{\beta\Delta_j(x)}} \right) \quad (31)$$

for Glauber dynamics, where  $\Delta_j = E(x \cdot z_j) - E(x)$ . Given the sparsity constraints of the function  $E$  and of the moves  $z_j \in \mathcal{M}$ , the quantity  $\Delta_j$  can actually be evaluated from a subset of qubits of the system register, namely  $\mathcal{N}_j = \{k | k \in \Omega_\ell, z_j \cap \Omega_\ell \neq \emptyset, \forall \ell\}$ . For single-spin flips on a  $(k, d)$ -local Hamiltonian,  $|\mathcal{N}_j| \leq kd$  by definition. For multi-spin flips  $z_j$ , we get  $|\mathcal{N}_j| \leq |z_j|kd$ .

Thus, the Boltzmann coin consists of a sequence of  $N$  conditional gates  $R_j$ , where  $R_j$  itself is a single-qubit rotation by an angle determined by the qubits in the set  $\mathcal{N}_j$ . Since each  $\mathcal{N}_j$  is of constant-bounded size, each  $R_j$  can be realized from a constant number of  $T$  gates, so the entire Boltzmann coin requires  $\mathcal{O}(N \log \frac{1}{\epsilon})$   $T$  gates, where  $\epsilon$  is the desired accuracy for the synthesis of single-qubits rotations. It is likely that a high precision is needed to ensure the detailed balance condition. We leave for future research the numerical investigation of how low the precision can be without causing significant errors.

Because all gates  $R_j$  act on the Coin register, they must be applied sequentially. An alternative consists in copying the Coin register in the conjugate basis of  $\sigma_y$ , i.e.  $|\pm i\rangle \rightarrow |\pm i\rangle^{\otimes N}$  since a sequence of rotations  $e^{i\theta_j \sigma_x}$  is equivalent to a tensor product of these rotations under this mapping. Moreover, any set of gates  $R_j$  with non-overlapping  $\mathcal{N}_j$  can be executed in parallel. Consequently, the total depth can be bounded by a constant at the expense of  $N$  additional qubits.

The complexity of the Boltzmann coin does scale exponentially with the sparsity parameters of the model however, namely as  $\mathcal{O}(\max_j 2^{|\mathcal{N}_j|})$ . A circuit that achieves  $R_j$  consists of a sequence of  $2^{|\mathcal{N}_j|}$  single-qubit rotations by an angle given by Eq. (30) or Eq. (31), conditioned on the bits in  $\mathcal{N}_j$  taking some fixed value. Each of these  $2^{|\mathcal{N}_j|}$  multi-controlled rotations require  $\mathcal{O}(|\mathcal{N}_j|)$  Toffoli gates along with  $\mathcal{O}(\log \frac{1}{\epsilon})$   $T$  gates, for an overall circuit depth of  $\mathcal{O}(2^{|\mathcal{N}_j|} |\mathcal{N}_j| \log \frac{1}{\epsilon})$  to realize  $R_j$ .

Perhaps a more efficient way to realize the Boltzmann coin uses quantum signal processing methods [11, 18–20]. This is a method to construct a unitary transformation  $S_2 = \sum_x f(e^{i\phi_x})|x\rangle\langle x|$  from a controlled version of  $S_1 = \sum_x e^{i\phi_x}|x\rangle\langle x|$ . In the current setting,  $S_1 = \sum_x e^{i\lambda\Delta_j(x)}|x\rangle\langle x|$  and we choose  $f(e^{i\lambda\Delta_j(x)}) = e^{i2\theta_{x,j}}$  where  $\theta_{x,j}$  is given at Eq. (30). Applying a Hadamard to the Coin qubit, followed by a controlled  $S_2$  with the Coin acting as control, and followed by a Hadamard on the Coin qubit again results in the transformation that we called  $R_j$  above and that builds up the Boltzmann coin transformation  $B$ .

Above, the constant  $\lambda$  is chosen in such a way that the argument of the exponential  $e^{i\lambda\Delta_j(x)}$  is restricted to some finite interval which does not span the entire unit circle, say in the range  $[-\pi/2, \pi/2]$ . The exponential can be further decomposed as a product

$$e^{i\lambda\Delta_j(x)} = \prod_{\ell: \Omega_\ell \cap z_j \neq \emptyset} \exp \left\{ i\lambda 2J_\ell \prod_{s \in \Omega_\ell} x_s \right\}. \quad (32)$$

Each of these factors is a rotation by an angle  $2J_\ell$ , whose sign is conditioned on the parity of the bits in  $\lambda\Omega_\ell$ . The parity bit can be computed using  $|\Omega_\ell|$  CNOTs, and the rotation is implemented using gate synthesis, with a  $T$ -gate count per transformation of  $\mathcal{O}(\log \frac{1}{\epsilon})$ , which is

dictated by the accuracy  $\epsilon$ . The complexity of quantum signal processing depends on the targeted accuracy. More precisely, it scales with the number of Fourier coefficients required to approximate the function  $f(e^{i\theta}) = \min(1, e^{-\theta\beta/\lambda})$  or  $g(e^{i\theta}) = \frac{1}{1+e^{\theta\beta/\lambda}}$  to some constant accuracy  $\epsilon$  on the domain  $\theta \in [-\pi/2, \pi/2]$ .

Quantum signal processing, or alternative methods, will offer an advantage on some models, when there are different couplings and a high number of body interactions for example. The scaling of these methods is case dependent. Indeed, it will highly depend on these couplings and the number of spin flips  $z_j$ .

### 3 Heuristic use

The Metropolis-Hastings algorithm is widely used heuristically to solve minimization problems using simulated annealing or related algorithms [15]. The objective function is the energy  $E(x)$ . Starting from a random configuration or an informed guess, the random walk is applied until some low-energy configuration  $x$  is reached. The parameter  $\beta$  can be varied in time, with an initial low value enabling large energy fluctuations to prevent the algorithm from getting trapped in local minimums, and large final value to reach a good (perhaps local) minimum.

In this section, we propose heuristic ways to use the quantum walk in the context of a minimization problem. We first recall the concept of total time to solution [24] which we use to benchmark and compare different heuristics. We then present two quantum heuristics which we compare using numerical simulations on small instances.

Since the purpose of our study is to compare a classical walk to its different quantum incarnations – as opposed to optimizing a classical walk – we will use a schedule with a linearly increasing value of  $\beta$  in time up to a fixed final value of  $\beta$  in our comparison and expect our conclusions to hold if an optimized  $\beta$  schedule was used instead in both the classical and the quantum walks.

#### 3.1 Total time to solution

When a random walk is used to minimize some function  $E(x)$ , the minimum  $x^*$  is only reached with some finite probability  $p$ . Starting from some distribution  $q(x)$  and applying the walk  $\mathcal{W}$

sequentially  $t$  times, the success probability is  $p(t) = (\mathcal{W}^t q)(x^*)$ . To boost this probability to some constant value  $1 - \delta$ , it is sufficient to repeat the procedure  $L = \frac{\log(1-\delta)}{\log(1-p(t))}$  times. The total time to solution is then defined as the duration of the walk  $t$  times the number of repetitions  $L$ ,

$$\text{TTS}(t) := t \frac{\log(1-\delta)}{\log(1-p(t))}. \quad (33)$$

There is a compromise to be reached between the duration of the walk  $t$  and the success probability  $p(t)$  – longer walks can reach a higher success probability and therefore be repeated fewer times, but increasing the duration  $t$  of the walk beyond a certain point has a negligible impact on its success probability  $p(t)$ . We thus define the minimum total time to solution as  $\min(\text{TTS}) = \min_t \text{TTS}(t)$ .

#### 3.2 Zeno with rewind

In Sec. 1.2, we explained how to prepare the eigenstate of  $U_{\mathcal{W}}$  with eigenvalue 1 using a sequence of walks  $\mathcal{W}^0, \mathcal{W}^1, \dots, \mathcal{W}^L = \mathcal{W}$ . In the setting of Metropolis-Hastings where  $\mathcal{W}$  is the walk with parameter  $\beta$ , a natural choice of  $\mathcal{W}^j$  is given by  $\beta^j = \frac{j}{L}\beta$ . An optimized  $\beta$  schedule is also possible, but for a systematic comparison with the classical walk, we choose this fixed schedule, whose only parameter is the number of steps  $L$ .

Let us revisit the argument of Sec. 1.2 to establish some notation. Define the binary projective measurement  $\{Q_j, Q_j^\perp\} := \{|\pi^j\rangle\langle\pi^j|, I - |\pi^j\rangle\langle\pi^j|\}$ . This binary measurement can be realized from  $\frac{1}{\delta_j}$  uses of  $U_{\mathcal{W}^j}$ , where  $\delta_j$  denotes the spectral gap of  $U_{\mathcal{W}^j}$ . Starting from the state  $|\pi^0\rangle$ , the Zeno algorithm consists in performing the sequence of binary measurements  $\{Q_j, Q_j^\perp\}$  in increasing value of  $j$ . The outcome  $Q_j$  on state  $|\pi^{j-1}\rangle$  yields state  $|\pi^j\rangle$  and occurs with probability  $F_j^2 := |\langle\pi^{j-1}|\pi^j\rangle|^2$ . The sequence of measurements succeeds if they all yield this outcome, which occurs with probability  $\prod_{j=1}^L F_j^2$  and requires  $\sum_{j=1}^L \frac{1}{\delta_j}$  applications of quantum walk operator. For the algorithm to be successful, the final measurement in the computational basis must also yield the optimal outcome  $x^*$ , which occurs with probability  $\pi^L(x^*)$ . Thus, the total time to

solution for an  $L$ -step algorithm is

$$\text{TTS}(L) = \frac{\log(1 - \delta)}{\log(1 - \pi^L(x^*) \prod_{j=1}^L F_j^2)} \sum_{j=1}^L \frac{1}{\delta_j}. \quad (34)$$

In the method outlined above, a measurement outcome  $Q_j^\perp$  requires a complete restart of the algorithm to  $\beta = 0$ . There exists an alternative to a complete restart which we call rewind. It was first described in the context of Zeno state preparation in Ref. [17], but originates from Refs. [22, 29]. It consists of iterating between the measurements  $\{Q_{j-1}, Q_{j-1}^\perp\}$  and  $\{Q_j, Q_j^\perp\}$  until the measurement  $Q_j$  is obtained. It can easily be shown that a transition between outcomes  $Q_{j-1} \leftrightarrow Q_j$  or  $Q_{j-1}^\perp \leftrightarrow Q_j^\perp$  is  $F_j^2$  while the probability of a transition between outcomes  $Q_{j-1} \perp \leftrightarrow Q_j$  or  $Q_{j-1}^\perp \leftrightarrow Q_j^\perp$  is  $1 - F_j^2$ . Given the cost  $\frac{1}{\delta_j}$  of each of these measurements, we obtain a simple recursion relation for the expected cost of a successful  $|\pi^{j-1}\rangle \rightarrow |\pi^j\rangle$  transition with rewind, and thus for the total time to solution for a  $L$ -step Zeno protocol with rewind. The minimal total time to solution is obtained by minimizing over  $L$ . In Ref. [17], it was found that rewinding yields substantial savings compared to the regular Zeno strategy for the preparation of quantum many-body ground states.

### 3.3 Unitary implementation

We propose another heuristic use of the quantum walk which does not use measurement. Starting from state  $|\pi^0\rangle$ , it consists in applying the quantum walk operators  $U_{\mathcal{W}^j}$  sequentially, resulting in the state

$$|\psi(L)\rangle = U_{\mathcal{W}^L} \dots U_{\mathcal{W}^2} U_{\mathcal{W}^1} |\pi^0\rangle, \quad (35)$$

and ending with a computational basis measurement. The algorithm is successful if a computational basis measurement yields the outcome  $x^*$  on state  $|\psi(L)\rangle$  (rewind could be used otherwise), so the total time to solution for the  $L$ -step algorithm is

$$\text{TTS}(L) = \frac{\log(1 - \delta)}{\log(1 - |\langle x^* | U_{\mathcal{W}^L} \dots U_{\mathcal{W}^2} U_{\mathcal{W}^1} |\pi^0\rangle|^2)}. \quad (36)$$

While we do not have a solid justification for this heuristic use, in Ref. [7], a protocol was proposed which used a similar sequence of unitaries, but where each unitary was applied a random

number of times. The motivation for these randomized transformations was to phase randomize in the eigenbasis of the instantaneous unitary operator. When the spectral gap of a unitary operator is  $\delta$  and that unitary is applied a random number of times in the interval  $[0, \frac{1}{\delta_j}]$ , then the relative phase between the eigenstate with eigenvalue 1 and the other eigenstates is randomized over the unit circle, thus mimicking the effect of a measurement (but with an unknown outcome). From this analogy, we could expect that the unitary implementation yields a minimal total time to solution roughly equal to the Zeno-based algorithm with no rewind. But as we will see in the next section, its behavior is much better than anticipated – this method is more efficient than the Zeno algorithm with rewind, which itself is more efficient than Zeno without rewind.

### 3.4 Numerical results

We have numerically benchmarked three heuristic algorithms: the classical walk with a variable-length linear interpolation between  $\beta = 0$  and  $\beta = 2$  and starting from a uniform distribution; the discrete, or Zeno-based adiabatic algorithm with rewind; and the unitary algorithm of the last subsection. The first system considered is a one dimensional Ising model. Figure 1 shows the quantum versus classical minimal total time to solution. The results clearly indicate a polynomial advantage of the quantum algorithms over the classical algorithm. Surprisingly, both quantum approaches show a similar improvement over the classical approach that exceed the expected quadratic speedup, with a power law fit of 0.42 using the unitary algorithm and 0.39 using the Zeno algorithm.

The second system considered is a sparse random Ising model: it has gaussian random couplings  $J_\ell$  of variance 1, and the interactions sets  $\Omega_\ell$  (c.f. Eq. (32)) consist of a random subset of  $3.5n$  of all the  $n(n-1)/2$  pairs of sites. Figure 2 shows quantum versus classical minimum total time to solution for a random ensemble of 100 systems of each sizes  $n = 4$  to 14. We observe that the unitary algorithm is consistently faster than the classical algorithm, with an average polynomial speedup of degree 0.75, less than the expected quadratic gain. Moreover, the different problem instances are all quite clustered around this average behavior, suggesting that the

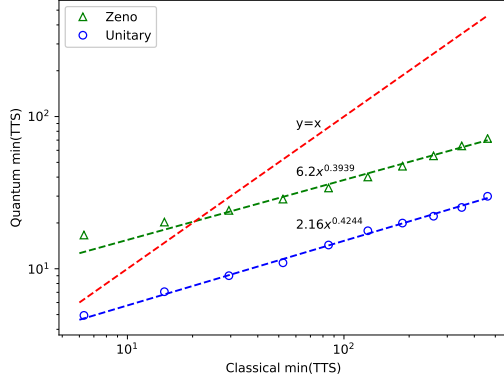


Figure 1: Quantum versus classical minimum total time to solution ( $\min(\text{TTS})$ ) for a one dimensional Ising model of length ranging from  $n = 3$  to 12 at  $\beta = 2$ . The line  $x = y$  is shown for reference of a quantum speedup.

quantum speedup is fairly general and consistent. In contrast the Zeno algorithm shows large fluctuations about its average, particularly on very small problem instances. The average polynomial speedup is of degree 0.92, far worse than the unitary algorithm. Overall, the results indicate a polynomial advantage of the quantum methods over the classical method, but these advantages are much less pronounced than for the 1D Ising model.

In both the one-dimensional and the random graph Ising model, the unitary quantum algorithm achieves very similar and sometimes superior scaling to the Zeno with rewind algorithm. This is surprising given the observed improvement obtained from rewind in Ref. [17] and our expectation that the unitary algorithm behaves essentially like Zeno without rewind.

## 4 Discussion

Our conclusion, and perhaps one of the key messages of this Article, is that even though the quantum walk is traditionally defined with the help of a walk oracle, its circuit implementation does not necessarily require it, and this can lead to substantial savings. In Appendix A, we discuss the difficulty of implementing the quantum walk unitary  $W$ . Appendix B presents an improved parallelized heuristic classical walk for discrete sparse optimization problems which could potentially lead to significant improvements on a quan-

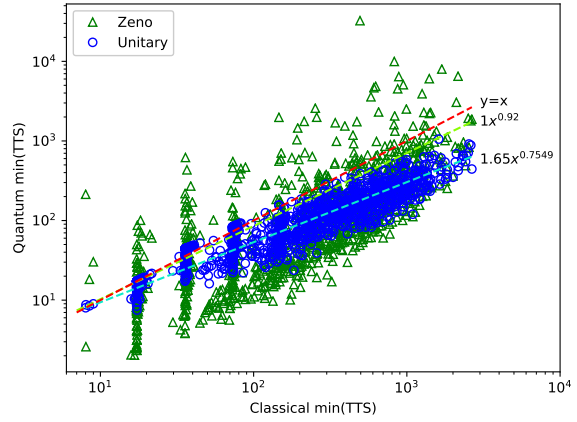


Figure 2: Quantum versus classical minimum total time to solution ( $\min(\text{TTS})$ ) for a random sparse Ising model at  $\beta = 2$ ,  $k = 2$  and  $d = 3.5n$ . 100 random problem instances are chosen for each size, ranging from  $n = 4$  to 14 .

tum computer. Unfortunately this walk is not reversible, which motivates further generalization of Szegedy’s quantization to include irreversible classical walks. In the rest of this section, we discuss the prospect of using the quantum walk to outperform a classical supercomputer.

We have proposed heuristic quantum algorithms based on the Szegedy walk for solving discrete optimization problems. Theoretical bounds show that the quantum algorithm can benefit from a quadratic speed-up ( $x^{0.5}$ ) over its classical counterpart. Our numerical simulations on small problem instances indicate a super-quadratic speed-up ( $\approx x^{0.42}$ ) for the Ising chain, see figure 1, and sub-quadratic speed-up ( $\approx x^{0.75}$ ) for random sparse Ising graphs, see figure 2. It remains an interesting question to understand more broadly what type of problems can benefit from what range of speed-up and why. With these crude estimates in hand we can already look into the achievability of a quantum speed-up on realistic devices.

We will compare performances to the special-purpose supercomputer “Janus” [13, 14] which consists of a massive parallel field-programmable gate array (FPGA). This system is capable of performing  $10^{12}$  Markov chain spin updates per second on a three-dimensional Ising spin glass of size  $n = 80^3$ . A calculation that lasts a bit less than a month will thus realize  $10^{18}$  Monte Carlo steps. On the one hand, assuming that the theoretically predicted quadratic speed-up holds and since the



numerics show a constant factor around 1, the quantum computer must realize at least  $10^9$  steps per month in order to keep up with the classical computer. This requires that a single step of the quantum walk be realized in a few milliseconds. On the other hand, the super-quadratic speed-up we have observed would allow almost a tenth of a second to realize a single quantum step, while the sub-quadratic speed-up would require that a single step be realized within 0.1 microseconds.

Taking the circuit depth reported in Table 1 as reference with  $d = 6$  for a three-dimensional lattice leads to a circuit depth of  $\log(80^3) \times 2^6 \approx 1000$ . To avoid harmful error accumulation, the gate synthesis accuracy  $\epsilon$  should be chosen as the inverse volume (circuit depth times the number of qubits) of the quantum circuit, roughly  $\epsilon^{-1} \approx 80^3 \times \log(80^3) \times 10^9 \approx 10^{16}$ , so on the order of  $4 \log \frac{1}{\epsilon} \approx 200$  logical  $T$  gates are required per fine-tuned rotation [5, 25], for a total logical circuit depth of 200,000. With these estimates, the three scenarios described above require logical gate speeds ranging from an unrealistically short 0.5 picoseconds (sub-quadratic speed-up), to an extremely challenging 1 nanosecond (quadratic speed-up), and allow 0.5 microseconds (super-quadratic speed-up).

We could instead compile the rotations offline and teleport them in the computation [8], which requires at least  $4 \log \frac{1}{\epsilon} \approx 200$  more qubits, but increases the time available for a logical gate by the same factor. Under this scenario, the time required for each logical gate would range from 0.1 nanoseconds (sub-quadratic speed-up), to 20 microseconds (quadratic speed-up), and to 1 milliseconds (super-quadratic speed-up). These estimates are summarized in Table 2. The latter is a realistic logical gate time for many qubit architectures, while there is no current path to achieve nanosecond logical gate times.

Given the above analysis, if a quantum computer is to offer a practical speed-up, we conclude that a better understanding of the class of problems for which heuristic super-quadratic speed-ups can be achieved is required, and that we need to optimize circuit implementations even further.

## 5 Acknowledgements

We thank Jeongwan Haah, Thomas Häner, Matt Hastings, Guang Hao Low and Guillaume Duclos-

Cianci for stimulating discussions. JL acknowledges support from the FRQNT programs of scholarships.

## References

- [1] Dorit Aharonov and Amnon Ta-Shma. Adiabatic quantum state generation and statistical zero knowledge. In *Proceedings of the thirty-fifth ACM symposium on Theory of computing - STOC '03*, page 20, New York, New York, USA, 2003. ACM Press. ISBN 1581136749. DOI: [10.1145/780542.780546](https://doi.org/10.1145/780542.780546).
- [2] Andris Ambainis. Quantum walk algorithm for element distinctness. In *Proceedings - Annual IEEE Symposium on Foundations of Computer Science, FOCS*, pages 22–31, 2004. DOI: [10.1109/focs.2004.54](https://doi.org/10.1109/focs.2004.54).
- [3] Francisco Barahona. On the computational complexity of Ising spin glass models. *Journal of Physics A: Mathematical and General*, 15:3241–3253, 1982. DOI: [10.1088/0305-4470/15/10/028](https://doi.org/10.1088/0305-4470/15/10/028).
- [4] Adriano Barenco, Charles H. Bennett, Richard Cleve, David P. Divincenzo, Norman Margolus, Peter Shor, Tycho Sleator, John A. Smolin, and Harald Weinfurter. Elementary gates for quantum computation. *Physical Review A*, 52(5):3457–3467, nov 1995. ISSN 10502947. DOI: [10.1103/PhysRevA.52.3457](https://doi.org/10.1103/PhysRevA.52.3457).
- [5] Alex Bocharov, Martin Roetteler, and Krysta M. Svore. Efficient synthesis of probabilistic quantum circuits with fallback. *Physical Review A - Atomic, Molecular, and Optical Physics*, 91(5):052317, may 2015. ISSN 10941622. DOI: [10.1103/PhysRevA.91.052317](https://doi.org/10.1103/PhysRevA.91.052317).
- [6] S. Boixo, E. Knill, and R. D. Somma. Fast quantum algorithms for traversing paths of eigenstates. may 2010. URL <https://arxiv.org/abs/1005.3034>.
- [7] Sergio Boixo, Emanuel Knill, and Rolando Somma. Eigenpath traversal by phase randomization. *Quantum Information and Computation*, 9(9&10):0833, 2009. URL <http://arxiv.org/abs/0903.1652>.
- [8] N Cody Jones, James D Whitfield, Peter L McMahon, Man-Hong Yung, Rodney Van Meter, Alán Aspuru-Guzik, and Yoshihisa Yamamoto. Faster quantum chemistry simu-

Quantum speedup	Synthesis online	Synthesis offline
Sub-quadratic $x^{0.75}$	0.5ps	0.1ns
Quadratic $x^{0.5}$	1ns	20 $\mu$ s
Super-quadratic $x^{0.42}$	0.5 $\mu$ s	1ms

Table 2: Logical gate time required to outperform a supercomputer capable of realizing  $10^{12}$  Monte Carlo updates per nanosecond in a computation that lasts one month. Arbitrary single-qubit rotations can be synthesized online or offline at an additional qubit cost.

- lation on fault-tolerant quantum computers. *New Journal of Physics*, 14(11):115023, nov 2012. ISSN 1367-2630. DOI: [10.1088/1367-2630/14/11/115023](https://doi.org/10.1088/1367-2630/14/11/115023).
- [9] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Michael Sipser. Quantum Computation by Adiabatic Evolution. jan 2000. URL <http://arxiv.org/abs/quant-ph/0001106>.
- [10] Roy J. Glauber. Time-dependent statistics of the Ising model. *Journal of Mathematical Physics*, 4(2):294–307, feb 1963. ISSN 00222488. DOI: [10.1063/1.1703954](https://doi.org/10.1063/1.1703954).
- [11] Jeongwan Haah. Product Decomposition of Periodic Functions in Quantum Signal Processing. jun 2018. DOI: [10.22331/q-2019-10-07-190](https://doi.org/10.22331/q-2019-10-07-190).
- [12] W K Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, apr 1970. ISSN 0006-3444. DOI: [10.1093/biomet/57.1.97](https://doi.org/10.1093/biomet/57.1.97).
- [13] Janus Collaboration, F. Belletti, M. Cottallo, A. Cruz, L. A. Fernández, A. Gordillo, M. Guidetti, A. Maiorano, F. Mantovani, E. Marinari, V. Martín-Mayor, A. Muñoz-Sudupe, D. Navarro, G. Parisi, S. Pérez-Gaviro, M. Rossi, J. J. Ruiz-Lorenzo, S. F. Schifano, D. Sciretti, A. Tarancón, R. Tripiccion, and J. L. Velasco. JANUS: an FPGA-based System for High Performance Scientific Computing. *Computing in Science & Engineering*, 11(1):48–58, 2009. DOI: [10.1109/MCSE.2009.11](https://doi.org/10.1109/MCSE.2009.11).
- [14] Janus Collaboration, M. Baity-Jesi, R. A. Banos, A. Cruz, L. A. Fernandez, J. M. Gil-Narvion, A. Gordillo-Guerrero, M. Guidetti, D. Iniguez, A. Maiorano, F. Mantovani, E. Marinari, V. Martin-Mayor, J. Monforte-Garcia, A. Munoz Sudupe, D. Navarro, G. Parisi, M. Pivanti, S. Perez-Gaviro, F. Ricci-Tersenghi, J. J. Ruiz-Lorenzo, S. F. Schifano, B. Seoane, A. Tarancon, P. Tellez, R. Tripiccion, and D. Yllanes. Reconfigurable computing for Monte Carlo simulations: results and prospects of the Janus project. *The European Physical Journal Special Topics*, 210(33), 2012. DOI: [10.1140/epjst/e2012-01636-9](https://doi.org/10.1140/epjst/e2012-01636-9).
- [15] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983. ISSN 00368075. DOI: [10.1126/science.220.4598.671](https://doi.org/10.1126/science.220.4598.671).
- [16] A. Yu. Kitaev. Quantum measurements and the Abelian Stabilizer Problem. nov 1995. URL <http://arxiv.org/abs/quant-ph/9511026>.
- [17] Jessica Lemieux, Guillaume Duclos-Cianci, David Sénéchal, and David Poulin. Resource estimate for quantum many-body ground state preparation on a quantum computer. 2020. URL <https://arxiv.org/abs/2006.04650>.
- [18] Guang Hao Low and Isaac L. Chuang. Hamiltonian Simulation by Qubitization. oct 2016. DOI: [10.22331/q-2019-07-12-163](https://doi.org/10.22331/q-2019-07-12-163).
- [19] Guang Hao Low and Isaac L. Chuang. Optimal Hamiltonian Simulation by Quantum Signal Processing. *Physical Review Letters*, 118(1):010501, jan 2017. ISSN 10797114. DOI: [10.1103/PhysRevLett.118.010501](https://doi.org/10.1103/PhysRevLett.118.010501).
- [20] Guang Hao Low, Theodore J. Yoder, and Isaac L. Chuang. Methodology of resonant equiangular composite quantum gates. *Physical Review X*, 6(4):041067, dec 2016. ISSN 21603308. DOI: [10.1103/PhysRevX.6.041067](https://doi.org/10.1103/PhysRevX.6.041067).
- [21] F. Magniez, A. Nayak, J. Roland, and M. Santha. Search via quantum walk. *SIAM Journal on Computing*, 40:142–164. DOI: [10.1137/090745854](https://doi.org/10.1137/090745854).
- [22] Chris Marriott and John Watrous. Quantum Arthur-Merlin games. In *Computational*

- Complexity*, volume 14, pages 122–152. Springer, jun 2005. DOI: [10.1007/s00037-005-0194-x](https://doi.org/10.1007/s00037-005-0194-x).
- [23] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, jun 1953. ISSN 00219606. DOI: [10.1063/1.1699114](https://doi.org/10.1063/1.1699114).
- [24] Troels F. Rønnow, Zhihui Wang, Joshua Job, Sergio Boixo, Sergei V. Isakov, David Wecker, John M. Martinis, Daniel A. Lidar, and Matthias Troyer. Defining and detecting quantum speedup. *Science*, 345(6195):420–424, jul 2014. ISSN 10959203. DOI: [10.1126/science.1252319](https://doi.org/10.1126/science.1252319).
- [25] Neil J Ross and Peter Selinger. Optimal ancilla-free Clifford+T approximation of Z-rotations. *Quantum Information and Computation*, 16(11&12):0901, 2016. URL <http://arxiv.org/abs/1403.2975>.
- [26] Terry Rudolph and Lov Grover. A 2 rebit gate universal for quantum computing. oct 2002. URL <https://arxiv.org/abs/quant-ph/0210187>.
- [27] R. D. Somma, S. Boixo, H. Barnum, and E. Knill. Quantum simulations of classical annealing processes. *Physical Review Letters*, 101(13):130504, sep 2008. ISSN 00319007. DOI: [10.1103/PhysRevLett.101.130504](https://doi.org/10.1103/PhysRevLett.101.130504).
- [28] Mario Szegedy. Quantum speed-up of Markov Chain based algorithms. In *Proceedings - Annual IEEE Symposium on Foundations of Computer Science, FOCS*, pages 32–41, 2004. DOI: [10.1109/focs.2004.53](https://doi.org/10.1109/focs.2004.53).
- [29] K. Temme, T. J. Osborne, K. G. Vollbrecht, D. Poulin, and F. Verstraete. Quantum Metropolis sampling. *Nature*, 471(7336):87–90, mar 2011. ISSN 00280836. DOI: [10.1038/nature09770](https://doi.org/10.1038/nature09770).
- [30] Marija Vucelja. Lifting—A nonreversible Markov chain Monte Carlo algorithm. *American Journal of Physics*, 84(12):958–968, dec 2016. ISSN 0002-9505. DOI: [10.1119/1.4961596](https://doi.org/10.1119/1.4961596).
- [31] Man-Hong Yung and Alán Aspuru-Guzik. A quantum-quantum Metropolis algorithm. *Proceedings of the National Academy of Sciences of the United States of America*, 109(3):754–9, jan 2012. ISSN 1091-6490. DOI: [10.1073/pnas.1111758109](https://doi.org/10.1073/pnas.1111758109).

## A Walk oracle

Our implementation of the walk operator does not make use of the walk unitary  $W$  of Eq. (2). Since the transition matrix elements  $\mathcal{W}_{xy}$  can be computed efficiently, we know that  $W$  can be implemented in polynomial time. But this requires costly arithmetics which would yield a substantially larger complexity than the approach presented above.

To see how this complexity arises, consider the following implementation of  $W$ , which uses much of the same elements as introduced above. The computer comprises two copies of the system register, which we now label Left and Right. As before, it also comprises a Move register and a Coin register. Begin with the Left register in state  $x$  and all other registers in state 0. Use the transformation  $V$  to prepare the state  $|x\rangle_L \otimes |0\rangle_R \otimes |f\rangle_M \otimes |0\rangle_C$ . Using  $n$  CNOTs, copy the state of the Left register onto the Right register, resulting in  $|x\rangle_L \otimes |x\rangle_R \otimes |f\rangle_M \otimes |0\rangle_C$ . Apply the move  $z_j$  proposed by the Move register to the Right register. If the Move register is encoded in unary representation as above, this requires  $\mathcal{O}(N)$  CNOTs, and results in the state

$$|x\rangle_L \otimes \sum_{j \in \mathcal{M}} \sqrt{f(z_j)} |x \cdot z_j\rangle_R \otimes |z_j\rangle_M \otimes |0\rangle_C. \quad (37)$$

Using a version of the Boltzmann coin transformation on the Left, Right and Coin register yields

$$\begin{aligned} & |x\rangle_L \sum_j \sqrt{f(z_j)} |x \cdot z_j\rangle_R |z_j\rangle_M \\ & \otimes \left( \sqrt{1 - A_{(x \cdot z_j)x}} |0\rangle + A_{(x \cdot z_j)x} |1\rangle \right)_C \end{aligned} \quad (38)$$

$$\begin{aligned} & = |x\rangle_L \sum_{y \neq x} \sqrt{\mathcal{W}_{yz}} |y\rangle_R |x \cdot y\rangle_M \\ & \otimes \left( \sqrt{A_{yx}^{-1} - 1} |0\rangle + |1\rangle \right)_C. \end{aligned} \quad (39)$$

At this point, we swap the Left and Right registers conditioned on the Coin qubit being in state 1, resulting in the state

$$\begin{aligned} & \sum_{y \neq x} \sqrt{\mathcal{W}_{yx}} |y\rangle_L |x\rangle_R |x \cdot y\rangle_M |1\rangle_C \\ & + \sum_{y \neq x} \sqrt{f(x \cdot y)(1 - A_{yx})} |x\rangle_L |y\rangle_R |x \cdot y\rangle_M |0\rangle_C. \end{aligned} \quad (40)$$

Finally, reset the move register to 0 using  $2N$  CNOTs with controls from the Left and Right registers. At this point, the move register is disentangled and discarded, resulting in the state

$$\begin{aligned} & \sum_{y \neq x} \sqrt{\mathcal{W}_{yx}} |y\rangle_L |x\rangle_R |1\rangle_C \\ & + \sum_{y \neq x} \sqrt{f(x \cdot y)(1 - A_{yx})} |x\rangle_L |y\rangle_R |0\rangle_C. \end{aligned} \quad (41)$$

The relative weights of the two branches are the same as the classical MCMC methods, which corresponds to an acceptance rate of approximately  $1/2$ .

This is quite similar to the state that would result from the quantum walk operator  $W$  of Eq. (2), save for one detail. When the acceptance register is in state 0, the state  $\sum_{y \neq x} \sqrt{f(x \cdot y)(1 - A_{yx})} |y\rangle_R$  of the right register needs to be mapped to the state  $\sqrt{\mathcal{W}_{xx}} |x\rangle_R$ . Such a rotation clearly depends on all the coefficients  $A_{yx}$ , and all implementations we could envision used arithmetic operations that compute  $A_{xy}$ .

## B Irreversible parallel walk

Note that the Boltzmann operator  $B$  has a total number of gates that scales with the system size  $n$ , even though it is used to implement a single step of the quantum walk and that on average, a single spin is modified per step of the walk. This contrasts with the classical walk where in a single step of  $\mathcal{W}$ , a spin transition  $x \rightarrow x \cdot z$  is chosen with probability  $f(z)$ , the acceptance probability is computed, and the move is either accepted or rejected. Each transition  $x \rightarrow x \cdot z$  typically involves only a few spins (one in the setting we are currently considering), so implementing such a transition in the classical walk does not require an extensive number of gates. The complexity in that case is actually dominated by the generation of a pseudo-random number selecting the location of the spin to be flipped. As a consequence, the quantum algorithm suffers an  $n$ -fold complexity increase compared to the quantum algorithm.

This motivates the construction of a modified classical walk for the lattice spin model which also affects every spin of the lattice, putting the classical and quantum walks on equal footing in terms of gate count. For simplicity, suppose that the set of moves  $z_i \in \mathcal{M}$  consist in single-spin flips. We define a *parallel classical walk* with transition matrix

$$\mathcal{W}_{yx} = \prod_{j=1}^N [qB_j(x)]^{(1-\frac{x_j \cdot y_j}{2})} [1 - qB_j(x)]^{(1+\frac{x_j \cdot y_j}{2})}, \quad (42)$$

where  $B_i(x) = \min\{1, e^{\beta[E(x) - E(x \cdot z_i)]}\}$  and  $z_i$  is the transition which consists of flipping the  $i$ th spin only, so only spin  $i$  differ in  $x$  and  $x \cdot z_i$ . The variable  $0 \leq q \leq 1$  is a tunable parameter of the walk. In other words, a single step of this walk can be decomposed into a sequence over spins  $i$ , and consists of flipping  $i$  with probability  $q$  and accepting the flip with probability  $B_i(x) = \min\{1, e^{\beta[E(x) - E(x \cdot z_i)]}\}$ . Importantly, even if the moves are applied sequentially, the acceptance probability  $B_i(x)$  is always evaluated relative to the state at the beginning of the step, even though other spins could have become flipped during the sequence.

If instead the acceptance probability was evaluated conditioned on the previously accepted moves – i.e.  $B_i(x) = \min\{1, e^{\beta[E(x \cdot z_i^{\text{tot}}) - E(x \cdot z_i)]}\}$  where  $z_i^{\text{tot}}$  is the total transition accumulated up to step  $i$  – then this acceptance probability would be the same as used in the Metropolis-Hastings algorithm. Note that for a *local* spin model with, e.g., nearest-neighbor interactions, the two acceptance probabilities only differ if a neighbor of site  $i$  has been flipped prior to attempting to flip spin  $i$ . Because a transition on each spin is proposed with probability  $q$ , the probability of having two neighboring spins flipped is  $\mathcal{O}(q^2)$ . Thus, we essentially expect a single step of this modified walk to behave like  $qn$  steps of the original Metropolis-Hastings walk, with a systematic error that scales like  $nq^2$ . Moreover, this systematic error is expected to decrease over time since once the walk settles in a low-energy configuration, very few spin transitions will turn out to be accepted, thus further decreasing the probability of a neighboring pair of spin flips.

To verify the above expectation, we have performed numerical simulations on an Ising model

$$H = \sum_{i,j} J_{i,j} x_i x_j$$

where  $J_{i,j}$  were randomly chosen from  $\{+1, -1\}$ . Results are shown on Fig. 3. What we observe is that, for an equal amount of computational resources, the parallelized walk outperforms the original walk. This is true both in terms in reaching a quick pseudo minimum configuration at short times and in terms of reaching the true minimum at longer times. Thus, while this parallelization was introduced to ease the quantization procedure, it appears to be of interest on its own.

In this case, the quantum walk unitary  $W$  can easily be applied. We first proceed as in the previous subsection and use CNOTs to copy the Right register onto the Left register, yielding state  $|x\rangle_L \otimes |x\rangle_R$ . Then, sequentially over all spins  $i$ , apply a rotation to spin  $i$  of the Left register conditioned on the state of the spin  $i$  and its neighbors on the Right register. This rotation transforms  $|x_i\rangle \rightarrow \sqrt{1 - qB_i(x)}|x_i\rangle + \sqrt{qB_i(x)}|\bar{x}_i\rangle$ . Note that the function  $B_i(x)$  only depends on the bits of  $x$  that are

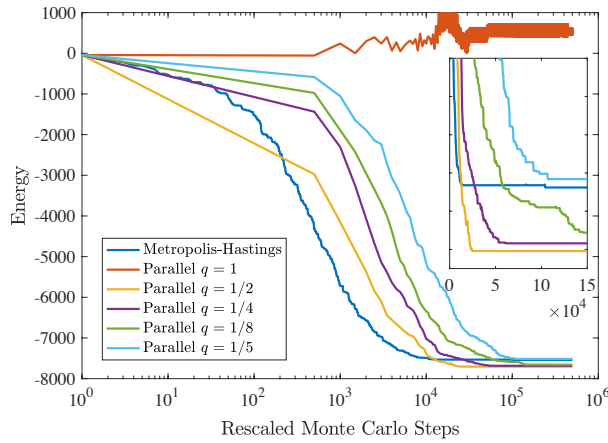


Figure 3: Energy above ground state of an Ising model on a complete graph with  $n = 500$  vertices with random binary couplings as a function of the number of Monte Carlo steps. Results are shown for regular Metropolis-Hastings walk and the parallelized walk with different values of  $q = 1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}$  and  $\frac{1}{16}$ . The temperature was set to  $\beta = 3$ , so the fixed point should be a low energy state. Since each step of the parallelized classical walk requires  $n = 500$  times as many gates as the original walk, the time label of the parallel walk has been multiplied by  $n$  so it adequately represent the number of computational resources. The parallel walk with  $q < 1$  outperforms the original walk at long times (see inset with first 150,000 steps) and achieves similar performances at short times as  $q$  approaches 1.

adjacent to site  $i$ , so this rotation acts on a constant number of spins so requires a constant number of gates. Thus, the cost of the classical and the quantum parallel walks have the same scaling in  $n$ . Combined to its observed advantages over the original classical walk, the parallel walk thus appears as the ideal version for a quantum implementation.

Unfortunately, the parallel walk is not reversible – it does not obey the detailed-balance condition Eq. (1). Thus, it is not directly suitable to quantization à la Szegedy. While quantization of non-reversible walks were considered in [21], they require an implementation of *time-reversed* Markov chain  $\mathcal{W}^*$  defined from  $\mathcal{W}$  and its fixed point  $\pi$  as

$$\mathcal{W}_{xy}^* \pi_x = \mathcal{W}_{yx} \pi_x. \quad (43)$$

Unfortunately, we do not know how to efficiently implement a quantum circuit for the time-reversed walk  $\mathcal{W}^*$ , so at present we are unable to quantize this parallel walk.

## Chapitre 4

# Algorithme adiabatique basé sur la réflexion

*Quantum mechanics is simpler than classical mechanics, if one leaves out the physics. In classical mechanics, we need to worry about whether the system is behaving like a wave or particle. While [in quantum mechanics], it is the same rules for waves and particles. Propagation will [behave] like waves and detection [...] like particles.*

— Grover, 2021 [39]

Nous avons pu le constater dans les deux derniers chapitres, la préparation d'état sur un ordinateur quantique peut être aussi bien une finalité qu'une sous-routine de l'algorithme principal. Comme il s'agit de trouver un état associé à un hamiltonien, il est facile de faire le parallèle avec l'algorithme de Grover. Ainsi, nous pouvons nous attendre au mieux à une accélération quadratique, du moins, pour une utilisation suivant les bornes analytiques d'un problème non structuré. Le fait est que les problèmes d'intérêts ont souvent un minimum de structure nous permettant par exemple de les exprimer en fonction d'un hamiltonien composé uniquement de termes à quelques corps. La formulation adéquate d'un problème sous forme d'évolution adiabatique pourrait permettre d'exploiter leur structure. Il n'est donc pas surprenant de voir que l'usage heuristique fonctionne bien de manière générale.

En nous inspirant des projets précédents, nous avons donc développé un nouvel algorithme : une évolution adiabatique basée sur la réflexion. Comme son nom l'indique, l'algorithme utilise des réflexions –plutôt que des mesures projectives– pour

calquer l'évolution. En ce sens, il s'agit d'un algorithme de Grover où l'espace succès évolue au cours du calcul.

## 4.1 Article

C'est moi qui ai eu l'idée innovante à l'origine de ce projet. La définition de la portée du projet a été faite par Pooya Ronagh – directeur du Hardware Innovation Lab chez IQBit – et moi-même. J'ai également fait les calculs et simulations nécessaires à l'entièreté des résultats qui ont été vérifiés par les co-auteurs. Ces derniers et moi-même avons tous participé à l'écriture de l'article.

Cet article a été soumis pour publication au journal *Quantum* et est disponible sur l'arXiv : Jessica Lemieux, Artur Scherer et Pooya Ronagh. Reflection-Based Adiabatic State Preparation, (2021), [arXiv:2111.05461](https://arxiv.org/abs/2111.05461).



# Reflection-Based Adiabatic State Preparation

Jessica Lemieux,<sup>1,2,3,\*</sup> Artur Scherer,<sup>4</sup> and Pooya Ronagh<sup>5,6,7,8</sup>

<sup>1</sup>*Département de Physique, Université de Sherbrooke, Sherbrooke, QC, Canada*

<sup>2</sup>*Institut Quantique, Université de Sherbrooke, Sherbrooke, QC, Canada*

<sup>3</sup>*1QBit Information Technologies (1QBit), Sherbrooke, QC, Canada*

<sup>4</sup>*1QBit, Waterloo, ON, Canada*

<sup>5</sup>*1QBit, Vancouver, BC, Canada*

<sup>6</sup>*Institute for Quantum Computing, University of Waterloo, Waterloo, ON, Canada*

<sup>7</sup>*Department of Physics & Astronomy, University of Waterloo, Waterloo, ON, Canada*

<sup>8</sup>*Perimeter Institute for Theoretical Physics, Waterloo, ON, Canada*

We propose a circuit-model quantum algorithm for eigenpath traversal that is based on a combination of concepts from Grover’s search and adiabatic quantum computation. Our algorithm deploys a sequence of reflections determined from eigenspaces of instantaneous Hamiltonians defined along an adiabatic schedule in order to prepare a ground state of a target problem Hamiltonian. We provide numerical evidence suggesting that, for combinatorial search problems, our algorithm can find a solution faster, on average, than Grover’s search. We demonstrate our findings by applying both algorithms to solving the NP-hard MAX-2SAT problem.

## I. INTRODUCTION

Grover’s search [15] is a famous quantum algorithm for unstructured search that offers a provable quadratic speed-up in comparison to exhaustive classical search. Variants of the algorithm have been studied since its invention 25 years ago [1, 9, 10, 12]. Grover’s algorithm was proven to be optimal in finding a marked element in an unstructured problem [30]. However, many industrially relevant real-world computational problems have additional structures, and exploiting them could result in potentially more-efficient heuristics that may outperform Grover’s search. Combinatorial and discrete optimization problems are examples of such structured problems. In our research, we consider the Boolean satisfiability problem as our working example. In particular, we use the NP-hard MAX-2SAT problem for our numerical studies.

We present a reflection-based adiabatic algorithm (RBA) based on a combination of concepts from Grover’s search and discrete adiabatic state preparation [2, 14, 19]. Our algorithm resembles eigenpath traversal navigated by the quantum Zeno effect [7, 8], but we replace projective measurements along the eigenpath with reflections to guide the evolution. In Sec. IID, we provide a discussion on the relationship between our research and previous work. The implementation of the RBA could follow any eigenpath traversal, such as along the instantaneous ground states of an adiabatic evolution. The algorithm’s sequence of reflections can be interpreted as slowly changing the marked state(s) when conducting a search using Grover’s algorithm.

To study the potential of the RBA, we benchmark its performance against that of Grover’s algorithm for small instances of the MAX-2SAT problem. We believe that

the RBA can also be successfully applied to solving optimization problems with non-classical objective functions. Examples of such problems include preparing the ground state of a generic Hamiltonian for fermionic systems, a problem known to be QMA-hard [17, 18, 26].

## II. ALGORITHM

The RBA uses a sequence of reflections ( $R_1, R_2, \dots, R_L$ ) defined by the ground states of a sequence of respective Hamiltonians. Let us denote the  $k$ -th ground state in this sequence by  $|G_k\rangle$ , where  $k \in \{1, \dots, L\}$ . The algorithm consists of the following main steps:

1. Prepare the initial state  $|\text{Init}\rangle$ .
2. For  $k = 1, \dots, L$ , apply the reflection  $R_k := \mathbb{1} - 2|G_k\rangle\langle G_k|$ .
3. Perform a measurement in the eigenbasis of the target problem Hamiltonian.

Depending on the type of problem, steps 1–3 may need to be repeated. We expand on the details of these steps in what follows.

### A. Intermediate Hamiltonians

To define the “good” subspaces through which the reflections are applied, we introduce intermediate Hamiltonians along an adiabatic path, although our approach allows more-generic paths. For simplicity, we use a linear interpolation between the starting Hamiltonian  $H_0$  and a problem Hamiltonian  $H_1$ :

$$H_w := (1 - w)H_0 + wH_1. \quad (1)$$

A discretization in time corresponds to a sequence of weights  $(w_1, w_2, \dots, w_L)$  that specify the instantaneous

\* Corresponding author: [jessica.lemieux@1qbit.com](mailto:jessica.lemieux@1qbit.com)

Hamiltonians  $(H_{w_1}, H_{w_2}, \dots, H_{w_L})$ . The ground states of these instantaneous Hamiltonians, respectively, characterize the corresponding good subspaces. The choice of the weights  $w_k$  could also be optimized classically by minimizing the expected energy of the resulting states, defining a hybrid quantum algorithm. These kinds of hybrid quantum–classical algorithms are often studied in the context of NISQ and variational quantum algorithms [13, 27]. The classical component of these hybrid schemes is a nontrivial optimization problem that is expected to scale poorly. For example, the barren plateau [16, 25] is the phenomenon of gradient norms decreasing exponentially fast, causing the need for exponentially many gradient estimations to be made with regard to problem size. Therefore, in Sec. III B we present the results of our study on the decay rate of the gradient norms for our hybrid scheme.

## B. Reflections

The implementation of reflections requires information about the eigenstates of the intermediate Hamiltonians, which we do not have. However, this information can be coherently acquired by performing quantum phase estimation (QPE) in the basis of the respective intermediate Hamiltonian, each time a reflection is deployed. In order to do this coherently without collapsing the state, the QPE step is followed by an energy value comparison with a classical threshold  $E^*$ . This energy threshold could either be stored in an additional quantum register or be “hard-coded” in the arithmetic needed for the comparison. The result of the comparison is stored in a flag qubit. In other words, instead of measuring the energy register, which would collapse the superposition state, we obtain a flag qubit entangled with the eigenstates, such that the qubit is in the state  $|1\rangle$  when the eigenstates correspond to an energy below the threshold  $E^*$ , and in the state  $|0\rangle$  otherwise. Effectively, this output corresponds to marking the state around which the reflection is performed. Recall that QPE is an algorithm for estimating the phases associated with the eigenvalues of a unitary operator. In our analysis, the unitaries are given by  $U_k = e^{iH_{w_k}}$ , which have the same eigenstates as  $H_{w_k}$ . The phases that correspond to the eigenvalues of  $H_{w_k}$  are denoted by  $E_{w_k}^j$  for every  $j$ -th eigenvalue. Thus, to ensure that QPE differentiates between all eigenstates, the Hamiltonians  $H_0$  and  $H_1$  must be normalized such that  $0 \leq E_{w_k}^j < 2\pi$  for all  $k$  and  $j$ . Note that this approach requires us to have lower and upper bounds for the energy spectrum.

In what follows, we denote the  $j$ -th eigenstate of  $H_{w_k}$  by  $|\psi_{w_k}^j\rangle$ . The reflection  $R_k$ , for  $k = 1, \dots, L$ , can then be implemented as follows.

1. Deploy QPE with  $U_k := e^{iH_{w_k}}$  and an additional register of size  $M$  used to hold the estimated energy

value:

$$\text{QPE } |0\rangle^M |\Psi\rangle = \sum_{j=0}^{2^n-1} \langle \psi_{w_k}^j | \Psi \rangle |E_{w_k}^j\rangle |\psi_{w_k}^j\rangle. \quad (2)$$

Here,  $|\Psi\rangle$  is an arbitrary state and  $|E_{w_k}^j\rangle$  represents the energy value corresponding to the eigenstate  $|\psi_{w_k}^j\rangle$  of  $H_{w_k}$ .

2. Apply an energy value comparison with a classical threshold  $E^*$ , which requires arithmetic. Append a single-qubit ancilla initialized in the computational state  $|0\rangle$ . If and only if  $E_{w_k}^j < E^*$ , flip the ancilla. If  $E^*$  is chosen such that  $E_{w_k}^0 < E^* \leq E_{w_k}^1$  (assuming the ordering<sup>1</sup>  $E_{w_k}^0 < E_{w_k}^1 \leq E_{w_k}^2 \leq \dots$ ), this results in the entangled state

$$\begin{aligned} & \langle G_k | \Psi \rangle |E_{w_k}^0\rangle |G_k\rangle |1\rangle \\ & + \sum_{j=1}^{2^n-1} \langle \psi_{w_k}^j | \Psi \rangle |E_{w_k}^j\rangle |\psi_{w_k}^j\rangle |0\rangle, \end{aligned} \quad (3)$$

where  $|G_k\rangle = |\psi_{w_k}^0\rangle$ .

3. Apply the Pauli- $Z$  gate to the flag ancilla, which results in a negative phase for the first term:

$$\begin{aligned} & - \langle G_k | \Psi \rangle |E_{w_k}^0\rangle |G_k\rangle |1\rangle \\ & + \sum_{j=1}^{2^n-1} \langle \psi_{w_k}^j | \Psi \rangle |E_{w_k}^j\rangle |\psi_{w_k}^j\rangle |0\rangle. \end{aligned} \quad (4)$$

4. Uncompute the registers for the energy comparison and QPE to yield the following:

$$- \langle G_k | \Psi \rangle |G_k\rangle + \sum_{j=1}^{2^n-1} \langle \psi_{w_k}^j | \Psi \rangle |\psi_{w_k}^j\rangle. \quad (5)$$

A quantum circuit diagram for these steps is shown in Fig. 1. This figure illustrates that the implementation cost of a reflection is not significantly different from that of a projective measurement. A projective measurement would terminate after QPE deployment (step 1) by measuring the register holding the energy value (cf. [8, 19]). To ensure that QPE is able to differentiate between the ground state and the first excited state, we need to estimate the phases  $E_{w_k}^j$  with a precision high enough to distinguish between  $E_{w_k}^0$  and  $E_{w_k}^1$ . In other words, the number of qubits required to hold the energy values obtained from QPE must scale as  $M \in \mathcal{O}(\lceil \log_2(1/\Delta_k) \rceil)$ , where  $\Delta_k := E_{w_k}^1 - E_{w_k}^0$  is the  $k$ -th energy gap. As QPE scales

<sup>1</sup> In the event of the ground states being degenerate, the threshold is chosen to be between the ground state energy and the next energy level.

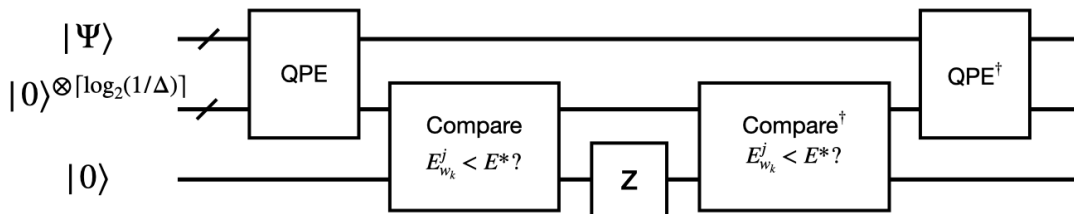


Figure 1. Quantum circuit for implementing a reflection

exponentially with respect to  $M$ , this leads to an overall scaling of  $\mathcal{O}(1/\Delta)$  in terms of a lower bound on the energy gaps,  $\Delta \leq \Delta_k$ , for all  $k = 0, \dots, L$ . With respect to the reflection, adding the energy comparison scales with the number of qubits as  $M \in \mathcal{O}(\lceil \log_2(1/\Delta_k) \rceil)$ . The uncomputation of the ancillary registers (necessary for reversible computation) doubles the cost. Thus, a projective measurement and a reflection have roughly the same scaling in terms of query complexity, which is  $\mathcal{O}(1/\Delta)$ .

There are multiple alternatives for the implementation of QPE. Normally, this algorithm requires Hamiltonian simulation of the unitary  $U_k = e^{iH_{w_k}}$ . This is typically performed using techniques based on Lie–Trotter product formulae [4], truncated Taylor series [5], quantum signal processing [23], or qubitization [24]. Two difficulties are usually encountered when implementing QPE by digitally simulating the operator  $U_k = e^{iH_{w_k}}$ : the error introduced by the Hamiltonian simulation and ambiguities in the phase. Using the framework of qubitization [24], these difficulties can be eliminated by replacing  $U_k$  with the qubiterate (see Refs. [6, 19, 28]). Since the energy spectrum is changed by qubitization, using the qubiterate will affect the query complexity of the reflections. The query complexity becomes  $\mathcal{O}(1/\arccos(1 - \Delta))$  instead of  $\mathcal{O}(1/\Delta)$  [19, 24]. There are also techniques that incorporate the linear combination of unitaries or oblivious amplitude amplification [11]. However, in view of the actual purpose of our work, we do not include these additional improvements. For simplicity, our analysis is based on the original approach of implementing QPE by simulating the unitary  $U_k$ .

### C. Connection to Grover’s Search Algorithm

Grover’s search [15], and the related amplitude amplification algorithm [10], use two reflections: one through the subspace spanned by the superposition state in the “good” subspace,  $|\text{Succ}\rangle$ , and another one through the subspace spanned by the initial state,  $|\text{Init}\rangle$ :

$$R_1^G = \mathbb{1} - 2|\text{Succ}\rangle\langle\text{Succ}| = e^{i\pi|\text{Succ}\rangle\langle\text{Succ}|}, \quad (6)$$

$$R_2^G = \mathbb{1} - 2|\text{Init}\rangle\langle\text{Init}| = e^{i\pi|\text{Init}\rangle\langle\text{Init}|}. \quad (7)$$

The algorithm repeatedly applies  $R_1$  followed by  $R_2$ , whose combined effect is called the Grover iteration. The key uncertainty is in figuring out when to stop

to achieve the highest probability of success. If  $|\langle\text{Succ}|\text{Init}\rangle| = \sin(\theta/2)$ , the optimal number of Grover iterations is  $n_{\text{it}}^{\text{opt}} = \lceil \frac{\pi}{2\theta} - \frac{1}{2} \rceil$ , in which case the algorithm outputs a solution state with a success probability equal to  $\sin^2[(n_{\text{it}}^{\text{opt}} + 1/2)\theta]$ . However, we generally do not know the overlap  $|\langle\text{Succ}|\text{Init}\rangle|$ . Since the probability of success is periodic in  $n_{\text{it}}$  in this scheme, exceeding  $n_{\text{it}}^{\text{opt}}$  results in its decrease. As for discrete adiabatic state preparation [7], in our scheme, increasing the number of intermediate steps will most likely increase the probability of success.

On the other hand, Grover’s algorithm can also be viewed as a special case of the RBA, where the weights  $w_k$  alternate between 0 and 1. Alternatively, imagine a perfectly defined reflection that maps the initial state to the desired target state in a single step. It can be shown that the gap of the intermediate Hamiltonian corresponding to such a reflection would be exactly  $\sin(\theta/2)$  (similar to the proof made by Roland and Cerf [29]). Thus, implementing this reflection using the steps described in the previous section would lead to the same scaling, because the gap determines the precision needed for QPE. Moreover, implementing a Trotter–Suzuki decomposition of this reflection, with  $R_1^G$  and  $R_2^G$  as the composing operators, would require a number of iterations that also scales with  $1/\sin(\theta/2)$ . Hence, we get the same scaling as for Grover’s algorithm.

### D. Connection to Eigenpath Traversal

The RBA performs a sequence of reflections to guide the evolution of a quantum state. This sequence of reflections can be found by a discretization of an adiabatic path, where the reflections are applied through the subspaces spanned by the ground states of the instantaneous Hamiltonians chosen. Previous studies have shown that the traversal of an eigenpath can be accomplished by a scheme resembling the quantum Zeno effect [7, 8, 19], preparing the target states with high fidelity by performing a sequence of projective measurements. Normally, the quantum Zeno effect involves frequently performing projective measurements that are all in the same basis, effecting a suppression of the system’s own dynamics and “localizing” its quantum state in one of the eigensubspaces of the measured observable. In the context of an adiabatic scheme, however, the measurement basis is

continually changed during the computation: in order to drag the system along an adiabatic evolution, the system state is projected onto the ground state of the instantaneous Hamiltonians of the corresponding adiabatic schedule. In the unlikely event of failure, the ground state can be recovered using a rewind procedure [19].

It should be noted that the algorithm by Boixo et al. [8] also uses reflections along the eigenpath (defined as a sequence of eigenstates  $|\psi_s\rangle$ , for  $0 \leq s \leq 1$ , of unitary operators  $U_s$  or Hamiltonians  $H_s$ ). However, in that scheme, each instantaneous reflection is controlled by an ancilla that is used to flag the desired eigenstate  $|G_k\rangle$  in that step. This flag qubit is subsequently measured, which is, in effect, the implementation of a binary projective measurement  $\{|G_k\rangle\langle G_k|, \mathbb{1} - |G_k\rangle\langle G_k|\}$ . In the event of failure, another reflection is used to introduce a significant overlap with  $|G_k\rangle$ . This is akin to the effect of performing a rewind procedure used by Lemieux et al. [19]. These operations are repeated until the flag qubit signals the  $|G_k\rangle$  has been successfully prepared. Hence, although using reflections, the overall progression of the algorithm [8] results in a quantum Zeno effect–like eigenpath traversal. In contrast, the RBA does not rely on intermediate projective measurements. Instead, it deploys consecutive reflections around the instantaneous eigenstates to traverse the eigenpath without forcing the evolving system state to “localize” in the instantaneous eigensubspace at each step along the schedule.

In what follows, we demonstrate that a reflection around an intermediate state gives a better probability of success than a binary projective measurement defined by the same state.

Let us denote the initial state and the final state by  $|I\rangle$  and  $|F\rangle$ , respectively. Beginning with  $|I\rangle$ , we aim to reach  $|F\rangle$  with as high a probability as possible. Furthermore, let us introduce a sequence of binary projective measurements,  $\{P_k, \bar{P}_k\}$ , where  $P_k := |k\rangle\langle k|$  and  $\bar{P}_k := \mathbb{1} - P_k$ , as well as a sequence of independent corresponding reflections,  $R_k = \mathbb{1} - 2|k\rangle\langle k|$ , for  $k \in \{1, \dots, L\}$ . Using the triangle inequality, given

$$\delta P := |\langle F|P_k|I\rangle| - |\langle F|I\rangle| \geq 0, \quad (8)$$

we have

$$\delta R := |\langle F|R_k|I\rangle| - |\langle F|I\rangle| \geq 2\delta P. \quad (9)$$

In other words, if an intermediate projective measurement improves the overlap between the initial and the final states, then the corresponding reflection will cause an even greater overlap. This also means that we can increase the probability of success by adding a reflection whenever adding the corresponding projective measurement is advantageous. Similarly, the use of a telescoping sum implies that

$$|\langle F|\prod_{k=1}^L R_k|I\rangle| - |\langle F|I\rangle| = \sum_{k=1}^L \delta R_k \geq 2 \sum_{k=1}^L \delta P_k, \quad (10)$$

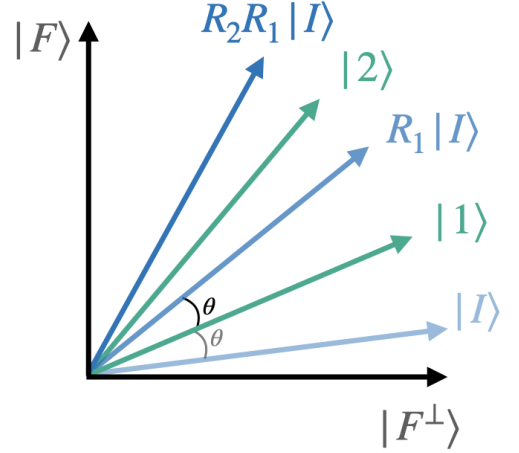


Figure 2. Visualization of the reflection-based preparation of the target state when adding reflections increases the overlap with the target state, as  $|\langle F|P_2R_1|I\rangle| > |\langle F|R_1|I\rangle|$

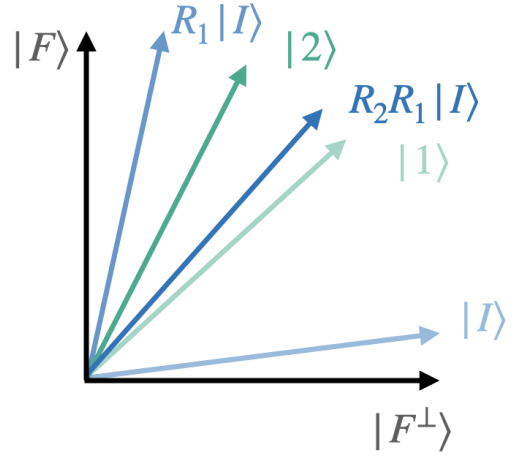


Figure 3. Visualization of the reflection-based preparation of the target state when adding a second reflection decreases the overlap with the target state, as  $|\langle F|P_2R_1|I\rangle| < |\langle F|R_1|I\rangle|$

where

$$\delta R_k := |\langle F|\prod_{j=1}^k R_j|I\rangle| - |\langle F|\prod_{j=1}^{k-1} R_j|I\rangle| \quad (11)$$

and

$$\delta P_k := |\langle F|P_k\prod_{j=1}^{k-1} R_j|I\rangle| - |\langle F|\prod_{j=1}^{k-1} R_j|I\rangle|. \quad (12)$$

We illustrate these concepts in Fig. 2 and in Fig. 3. Figure 2 shows when the condition needed to improve the probability of success is satisfied. Figure 3 illustrates the case when, even if making two projective measurements would increase the probability of success as compared to just one, the condition for increasing the probability

of success by adding a second reflection is not satisfied. Note that in the latter case, performing one reflection is actually better than making two projective measurements to increase the overlap with the target state. For details on the assumption that intermediate projective measurements will result in an increase in the overlap, see the work of Aharonov and Ta-Shma [3] and Boixo et al. [7].

### III. PERFORMANCE ANALYSIS OF THE RBA

To study the RBA, we compare its performance to Grover's search. We use the NP-hard MAX-2SAT problem as the reference problem.

#### A. Application to the MAX-2SAT Problem

In its conjunctive normal form, each MAX-2SAT problem instance can be expressed as the Boolean formula

$$\bigwedge_{c \in \mathcal{C}} (\ell_{c_1} \vee \ell_{c_2}), \quad (13)$$

where  $\mathcal{C}$  denotes the set of all clauses composing the formula. Here,

$$\ell_{c_1}, \ell_{c_2} \in \{v_0, \neg v_0, \dots, v_{n-1}, \neg v_{n-1}\} \quad (14)$$

are a pair of distinct literals specifying a clause  $c \in \mathcal{C}$ , and  $v_k \in \{\text{TRUE}, \text{FALSE}\}$ , for  $k = 0, 1, \dots, n-1$ , are  $n$  Boolean variables. Notice that  $c_1$  and  $c_2$  serve as labels for the two literals making up the clause  $c$ . For instance, for the clause  $c = (\neg v_i \vee v_j)$ ,  $c_1 = i$  and  $c_2 = j$ . The problem consists in the task of determining the maximum number of clauses  $c \in \mathcal{C}$  that can be simultaneously satisfied by an assignment to the Boolean variables.

The MAX-2SAT problem can be recast as the problem of finding the ground state of the 2-local Hamiltonian

$$H_c := \sum_{c \in \mathcal{C}} \left( (-1)^{\nu(\ell_{c_1})} (-1)^{\nu(\ell_{c_2})} Z_{c_1} Z_{c_2} + (-1)^{\nu(\ell_{c_1})} Z_{c_1} + (-1)^{\nu(\ell_{c_2})} Z_{c_2} \right), \quad (15)$$

where  $Z_k$  represents the Pauli operator pertaining to variable  $v_k$ ,  $\nu(\ell) = 0$  if the literal  $\ell$  is not negated, and  $\nu(\ell) = 1$  if it is. The mapping of the MAX-2SAT problem to the Hamiltonian is such that the eigenvalues  $+1$  and  $-1$  of the Pauli- $Z_k$  operator correspond to the Boolean values FALSE and TRUE of the variable  $v_k$ , respectively. The energy spectrum of this Hamiltonian is such that, every time a clause is not satisfied for a given variable assignment, there is a corresponding energy penalty of  $+3$ , and if a clause is satisfied, its energy contribution is  $-1$ . Therefore, the ground states of this Hamiltonian correspond to the variable assignments that

satisfy the maximum number of clauses. By shifting the energy spectrum by  $|\mathcal{C}|$ , the ground state energy would be equal to zero if the problem is satisfiable, and higher if it is not. This way, at any iteration, if the outcome of measuring the energy is zero, it is certain that the problem is satisfiable. Also, note that the entire spectrum is upper bounded by  $4|\mathcal{C}|$ .

In our analysis, the initial state is chosen to be the uniform superposition

$$|+\rangle^{\otimes n} := \text{HAD}^{\otimes n} |0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |k\rangle, \quad (16)$$

which can be easily generated by applying a Hadamard gate HAD to each qubit. This state is the ground state of the transverse field Hamiltonian

$$H_{\text{trans}} := - \sum_{k=0}^{n-1} X_k, \quad (17)$$

which is commonly chosen as the initial Hamiltonian of the adiabatic schedule, where  $X_k$  is the Pauli- $X$  operator corresponding to  $v_k$ .

#### B. Numerical Study

Our numerical benchmark includes simulations of the success probability of the RBA and a cost analysis in comparison to Grover's search in solving small MAX-2SAT problem instances. Our results pertain to classically optimized sets of reflection points and the ideal choice of scaling parameters for  $H_0$  and  $H_1$ . Note that, in practice, to optimize reflection points along the path, we would need to sample the final energy and use a classical optimizer to find a (sub-)optimal schedule for the reflections.

Our numerical simulations use the following steps.

1. Perform the ideal normalization as follows:

$$\begin{aligned} H_0 &:= 2\pi(H_{\text{trans}} - E_{\text{trans}}^0 \mathbb{1}) / (E_{\text{trans}}^{2^n-1} - E_{\text{trans}}^0), \\ H_1 &:= 2\pi(H_c - E_c^0 \mathbb{1}) / (E_c^{2^n-1} - E_c^0), \end{aligned} \quad (18)$$

where  $E_{\text{trans}}^j$  and  $E_c^j$  for  $j = 0, \dots, (2^n - 1)$  are the eigenvalues of energies of the Hamiltonian  $H_{\text{trans}}$  and  $H_c$ , respectively.

2. Set the total number of reflection points  $L = 1$ .
3. Use the Nelder-Mead optimization protocol to find a set  $\{w_1, \dots, w_L\}$  of reflection points (by minimizing the probability of failure), assuming energy thresholds between  $E_{w_k}^0$  and  $E_{w_k}^1$  for all  $k = 1, \dots, L$ , and compute the time to solution (TTS).
4. Repeat the previous step by performing  $L = 2, 3, 4, \dots$  reflections until a minimum TTS is observed.

In practice, we do not have access to  $E_{w_k}^0$  and  $E_{w_k}^1$  in step 3. However, we may use the known values of both  $E_0^0$  and  $E_0^1$ , and an upper bound for  $E_1^0$  determined using other techniques (e.g., polynomial-time approximation schemes [21] or local heuristic search methods [22]), to construct a concave threshold function with respect to  $w$  and tune the concavity of the function to estimate tight upper bounds for  $E_{w_k}^0$  for all reflection points  $w_k$ . We also note that the normalization in step 1 is considered “ideal” since we do not have access to  $E_1^0$  or  $E_1^{2^n-1}$ . However, we can use  $4|\mathcal{C}|$ , or another heuristically found bound, as an upper bound for  $(E_1^{2^n-1} - E_1^0)$ .

For small problem instances with the number of variables  $n \in \{5, \dots, 13\}$ , we compute the probability of success  $p_s$  (i.e., the modulus squared of the overlap between the final computational state and the target state), given the above choices of threshold, reflection points, and normalization. The TTS is then given by

$$\text{TTS}_{\text{RBA}} = \frac{\log \epsilon}{\log(1 - p_s)} \sum_{k=0}^L \frac{2r}{\Delta_k}, \quad (19)$$

where  $\Delta_k = E_{w_k}^1 - E_{w_k}^0$  is the normalized instantaneous energy gap,  $r := |\mathcal{C}|/n$ , and  $\epsilon$  is the overall allowable probability of failure. In our simulations, we set  $\epsilon = 0.1$ .

In Eq. (19), the cost of each trial of the algorithm is  $\sum_{k=0}^L \frac{2r}{\Delta_k}$ , as QPE requires a number of queries to Hamiltonian simulation unitaries that scales with the inverse of the precision needed to distinguish between the energies of the first excited state and the ground state. Moreover, each unitary has a gate depth that scales with the ratio  $r$ .

The TTS for Grover’s algorithm is given by

$$\begin{aligned} \text{TTS}_{\text{Grover}} &= \frac{\log \epsilon}{\log(1 - p_s)} \sum_{k=1}^{n_{it}} \frac{2r}{\Delta_T} \\ &= \frac{\log \epsilon}{\log(1 - p_s)} \frac{2n_{it}r}{\Delta_T}, \end{aligned} \quad (20)$$

where  $\Delta_T = 2\pi(E_C^1 - E_C^0)/(E_C^{2^n-1} - E_C^0)$  is the normalized gap pertaining to the target Hamiltonian. Note that we ignore the additional cost of  $R_1^G$ .

We generate 20 distinct random instances for each problem size and each  $r \in \{4, 6, 8\}$  (with the exception of the combination  $n = 5$  and  $r = 8$ , in which case there is only one possible instance), and compute the corresponding TTS values. We plot the results we obtain for the RBA against those we obtain for Grover’s search in Fig. 4 and in Fig. 5, where we can see a trend indicating a speed-up relative to Grover’s search. The TTS depends on both  $n$  and  $r$ . The impact of both of these parameters is shown in colour, with the colour bars indicating the number of variables in Fig. 4 and the ratio in Fig. 5.

These figures illustrate that, for small problem instances, Grover’s algorithm performs better, but, as the problem becomes harder, the RBA tends to outperform

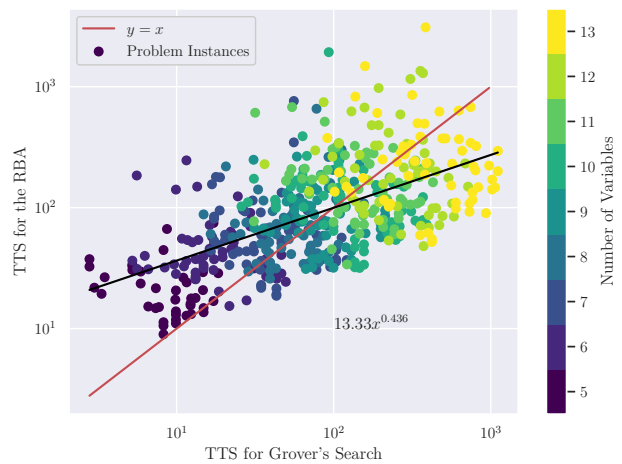


Figure 4. Improvement in TTS for the RBA compared to Grover’s search. The colours indicate the number of variables used for each problem instance.

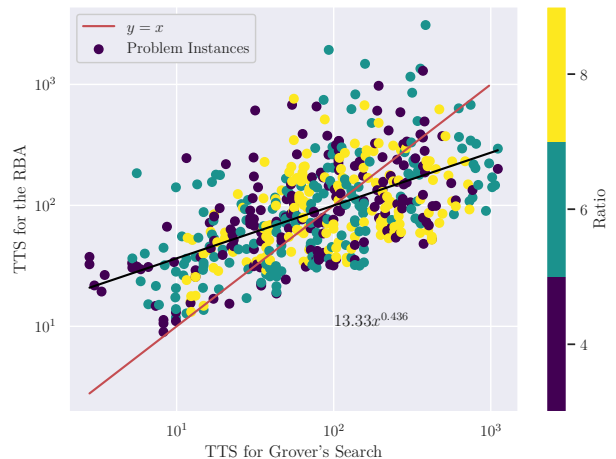


Figure 5. Improvement in TTS for the RBA compared to Grover’s search. The colours indicate the ratio used for each problem instance.

Grover’s algorithm. The reason is that the number of iterations required for our algorithm to reach a high probability of success grows more slowly with the system size than in Grover’s search, as shown in Fig. 6, but each reflection has a higher cost (due to there being a smaller gap for the instantaneous Hamiltonian in comparison to  $H_0$  and  $H_1$ ).

Figure 6 shows the median values of the number of iterations required for both algorithms to reach a target probability of failure below 0.2. As expected, for Grover’s algorithm, we observe an exponential dependence of the number of iterations on the number of variables. For the RBA, the nature of this relationship is not readily apparent. As the number of iterations is discrete and we only include simulations for problem instances with

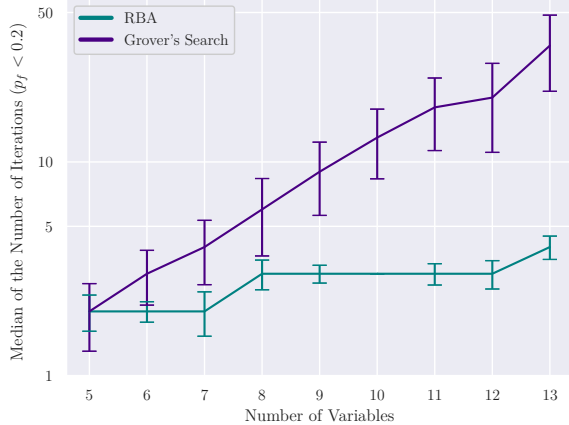


Figure 6. Median of the number of iterations versus the number variables for the RBA and Grover's search that is needed to obtain a probability of failure below a value of 0.2

up to 13 variables, it is hard to extract a clear trend. However, we observe a significant decrease in the number of iterations as compared to Grover's search.

The TTS scaling of the RBA could potentially be reduced by using the method of qubitization [6, 19, 24, 28]. Indeed, qubitization would change the overlap between successive states as well as the gap from  $\Delta$  to  $\arccos(1 - \frac{\Delta}{2\pi})$ , which has a greater impact for smaller gap values [19]. This could potentially result in a further improvement of the RBA over Grover's search.

For a hybrid approach, in which we optimize the choice of each  $w_k$  using a classical optimizer, we study the phenomenon of the barren plateau. To do so, we compute the variance  $\text{var}\left(\frac{\partial E}{\partial w_i}\right)$  when  $L = 2$ . We consider the same problem instances as earlier. To sample the variation in energy based on the positions of the reflections, we take 5000 random points uniformly distributed from within the intervals  $(\frac{1}{6}, \frac{1}{2})$  and  $(\frac{1}{2}, \frac{5}{6})$ . In Fig. 7, we observe an exponential decay with a rate of 0.08 for the median values of gradient variances. This is in contrast to the rate of 0.685 obtained for the random parameterized quantum circuits studied by McClean et al. [25]. The error bars indicate the standard deviation.

Figure 8 shows the TTS ratio of the non-optimized RBA (i.e., using equidistant reflection points between 0 and 1) over one that optimizes the reflection points versus the number of reflections (i.e., the advantage offered by the classical optimizer). We observe that not performing the optimization adds, on average, a linear factor with the number of reflections. Notice that, in some cases, the non-optimized version has a smaller TTS. This is because we minimize the probability of failure and not the TTS. Indeed, for some cases, the smaller probabilities of success exist alongside larger gaps, which can result in a smaller TTS. Figure 9 shows the TTS for the non-optimized RBA compared to the TTS for Grover's search.

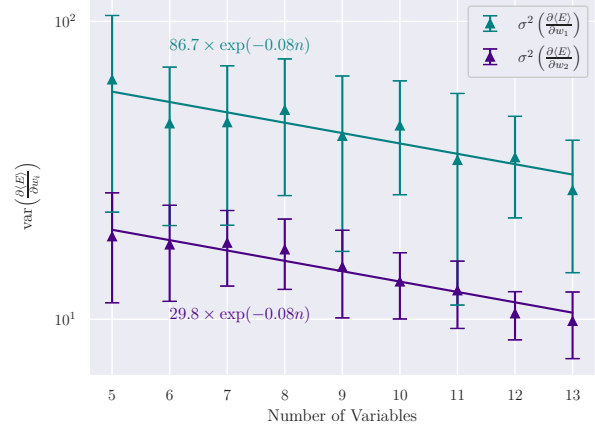


Figure 7. Variance of the partial derivative of the expected energy versus the number of variables for the RBA, demonstrating an exponential decay associated with the existence of a barren plateau for a schedule with two reflections

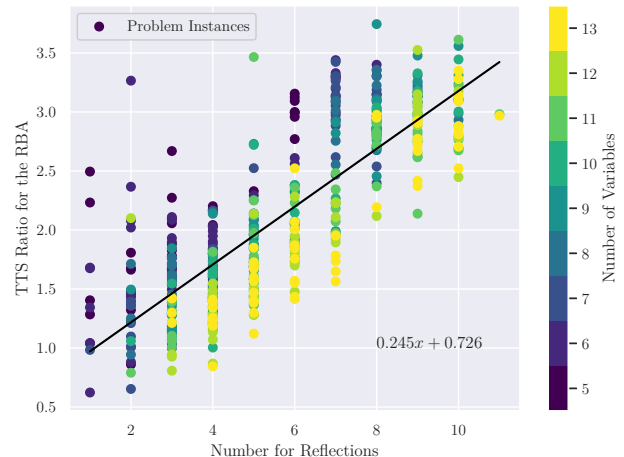


Figure 8. Time-to-solution ratio of the non-optimized RBA over one that is optimized versus the number of reflections. The non-optimized RBA uses a schedule with equidistant reflection points. The optimized RBA uses a schedule that minimizes the probability of failure. A decrease in the TTS resulting from optimizing the positions of the reflections can be seen. The colours indicate the number of variables used for each problem instance.

We observe that, even with a schedule with equidistant reflection points, the RBA offers a speed-up over Grover's search. Figure 9 is equivalent to Fig. 4, except that it does not include the optimization of the reflection points.

Finally, we empirically study the impact of having an energy threshold between  $E_{w_k}^1$  and  $E_{w_k}^2$  and observe that it does not result in a significant increase in the TTS. In fact, in some cases, it results in a decrease in the TTS, especially for instances that exhibit a closing energy gap toward the end of the adiabatic path. This is due to

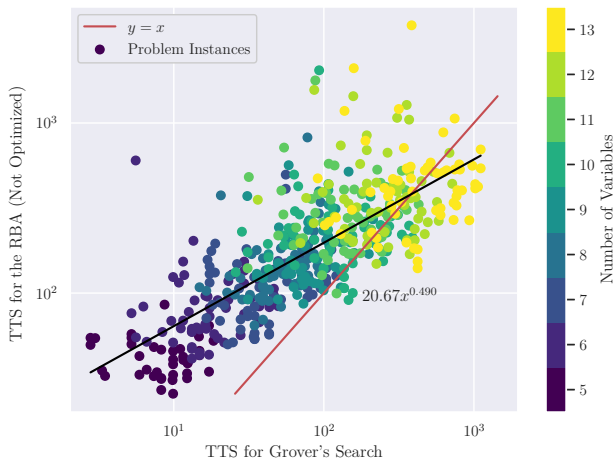


Figure 9. Time to solution for the non-optimized RBA versus the TTS for Grover’s search. Optimization of the reflection points is not performed; the figure is otherwise equivalent to Fig. 4. The non-optimized RBA uses a schedule with equidistant reflection points. The colours indicate the number of variables used for each problem instance.

there being a reduced-precision requirement needed to implement QPE. Indeed, in this case, the precision requirement scales inversely with  $(E_{w_k}^2 - E_{w_k}^0)$  instead of with  $(E_{w_k}^1 - E_{w_k}^0)$ . Thus, we expect that the RBA is not significantly sensitive to optimal choices of energy thresholds.

#### IV. CONCLUSION

We have presented a new circuit-model quantum algorithm for preparing the ground state of a target problem Hamiltonian. The reflection-based adiabatic algorithm (RBA) is based on a combination of concepts from Grover’s search and adiabatic quantum computation. We have shown that the resource requirements for the RBA to reach the target state with a high probability of success scale favourably compared to Grover’s search in solving MAX-2SAT problem instances. We used the time-to-solution metric (TTS), that is, the time needed to find a solution for a particular problem instance with high confidence (e.g., a rate of 90%). Although our algorithm has a higher implementation cost per reflection, as the probability of success increases faster with the number of reflections than it does in Grover’s search, the TTS scales better.

The complexity analysis made by Boixo et al. [8] suggests that path traversal algorithms could outperform Grover’s algorithm in solving search and optimization problems. In the present work, we have provided numerical evidence supporting this claim. Although the RBA does not employ projective measurements but instead uses reflections to traverse the eigenpath, the inequali-

ties from Sec. IID suggest that its query complexity is not worse than given by Boixo et al. [8].

Previous work on discrete adiabatic state preparation using continual instantaneous projective measurements [19] incorporated techniques such as qubitization [24] and a rewind procedure. We expect that qubitization would also reduce the implementation cost of the reflections in the RBA. However, a rewind procedure is not helpful in a unitary scheme (i.e., without projective measurements); thus, we have not employed such a procedure in our implementation, which benefits from full quantum parallelism. Indeed, it is not only the ground states along an adiabatic path that contribute to the probability amplitude of obtaining the desired target state, but so do the excited states. The contribution of the excited states becomes especially important in schemes where the gap closes toward the end of the adiabatic schedule, which is the case for degenerate instances of MAX- $k$ SAT problems. Moreover, in real-world experiments, the measurements required in measurement-based adiabatic schemes are generally slower and much more prone to suffering from errors than are unitary quantum gates. In light of our findings, we also believe that the heuristic use of the walk operator by Lemieux et al. [20] works well without employing phase randomization [7], because the walk operator consists of reflections.

In looking toward potential implementations of a hybrid RBA scheme supplemented by a classical optimization loop, we have investigated the barren plateau phenomenon. Our simulations indicate that there is a small exponential decay associated with the existence of a barren plateau, with a rate of 0.08. However, the decrease in the TTS resulting from optimizing the positions of reflections along an adiabatic path scales linearly with the number of reflections. Since the number of reflections employed in the RBA scales more slowly than that in Grover’s search, we expect that, even without optimization, the RBA algorithm will provide a greater speed-up, on average, than Grover’s search.

#### ACKNOWLEDGEMENT

The authors thank our editor, Marko Bucyk, for his careful review and editing this manuscript, Simon Verret and Gili Rosenberg for helpful discussions. P. R. additionally acknowledges the financial support of Mike and Ophelia Lazaridis, and Innovation, Science and Economic Development Canada.



- 
- [1] Scott Aaronson and Patrick Rall. Quantum Approximate Counting, Simplified. In *Symposium on Simplicity in Algorithms*, pages 24–32. SIAM, 2020. doi:10.1137/1.9781611976014.5.
- [2] Dorit Aharonov and Amnon Ta-Shma. Adiabatic quantum state generation and statistical zero knowledge. In *STOC '03: Proceedings of the thirty-fifth annual ACM symposium on Theory of Computing*, pages 20–29, 2003. doi:10.1145/780542.780546.
- [3] Dorit Aharonov and Amnon Ta-Shma. Adiabatic Quantum State Generation. *SIAM Journal on Computing*, 37(1):47–82, 2007. doi:10.1137/060648829.
- [4] Dominic W. Berry, Graeme Ahokas, Richard Cleve, and Barry C. Sanders. Efficient Quantum Algorithms for Simulating Sparse Hamiltonians. *Communications in Mathematical Physics*, 270:359–371, 2007. doi:10.1007/s00220-006-0150-x.
- [5] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma. Simulating Hamiltonian Dynamics with a Truncated Taylor Series. *Physical Review Letters*, 114:090502, Mar 2015. URL: <https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.114.090502>, doi:10.1103/PhysRevLett.114.090502.
- [6] Dominic W. Berry, Mária Kieferová, Artur Scherer, Yuval R. Sanders, Guang Hao Low, Nathan Wiebe, Craig Gidney, and Ryan Babbush. Improved techniques for preparing eigenstates of fermionic Hamiltonians. *npj Quantum Information*, 4:1(22):10501, 2018. doi:10.1038/s41534-018-0071-5.
- [7] Sergio Boixo, Emanuel Knill, and Rolando D. Somma. Eigenpath traversal by phase randomization. *Quantum Information and Computation*, 9(9&10):833–855, 2009. doi:10.26421/QIC9.9-10-7.
- [8] Sergio Boixo, Emanuel Knill, and Rolando D. Somma. Fast quantum algorithms for traversing paths of eigenstates. *arXiv:1005.3034*, 2010. URL: <https://arxiv.org/abs/1005.3034>.
- [9] Michel Boyer, Gilles Brassard, Peter Høyer, and Alain Tapp. Tight Bounds on Quantum Searching. *Fortschritte der Physik: Progress of Physics*, 46(4-5):493–505, 1998. doi:10.1002/(SICI)1521-3978(199806)46:4/5<493::AID-PROP493>3.0.CO;2-P.
- [10] Gilles Brassard, Peter Høyer, Michele Mosca, and Alain Tapp. Quantum Amplitude Amplification and Estimation. *Contemporary Mathematics*, 305:53–74, 2002. URL: <http://dx.doi.org/10.1090/conm/305>, doi:10.1090/conm/305/05215.
- [11] Anirban Narayan Chowdhury, Yigit Subasi, and Rolando D. Somma. Improved implementation of reflection operators. *arXiv:1803.02466*, 2018. URL: <https://arxiv.org/abs/1803.02466>.
- [12] Christoph Dürr and Peter Høyer. A Quantum Algorithm for Finding the Minimum. *arXiv:quant-ph/9607014*, 1996. URL: <https://arxiv.org/abs/quant-ph/9607014>.
- [13] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A Quantum Approximate Optimization Algorithm. *arXiv:1411.4028*, 2014. URL: <https://arxiv.org/abs/1411.4028>.
- [14] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, Joshua Lapan, Andrew Lundgren, and Daniel Preda. A Quantum Adiabatic Evolution Algorithm Applied to Random Instances of an NP-Complete Problem. *Science*, 292(5516):472–475, 2001. doi:10.1126/science.1057726.
- [15] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *STOC '96: Proceedings of the twenty-eighth annual ACM symposium on Theory of Computing*, pages 212–219, 1996. doi:10.1145/237814.237866.
- [16] Zoë Holmes, Kunal Sharma, M. Cerezo, and Patrick J. Coles. Connecting ansatz expressibility to gradient magnitudes and barren plateaus. *arXiv:2101.02138*, 2021. URL: <https://arxiv.org/abs/2101.02138>.
- [17] Julia Kempe, Alexei Kitaev, and Oded Regev. The Complexity of the Local Hamiltonian Problem. *Siam Journal of Computing*, 35(5):1070–1097, 2006. doi:10.1007/978-3-540-30538-5\_31.
- [18] Alexei Yu Kitaev, Alexander Shen, and Mikhail N. Vyalyi. *Classical and Quantum Computation*, volume 47. American Mathematical Society, 2002. URL: <http://dx.doi.org/10.1090/gsm/047>, doi:10.1090/gsm/047.
- [19] Jessica Lemieux, Guillaume Duclos-Cianci, David Sénéchal, and David Poulin. Resource estimate for quantum many-body ground-state preparation on a quantum computer. *Physical Review A*, 103:052408, May 2021. URL: <https://link.aps.org/doi/10.1103/PhysRevA.103.052408>, doi:10.1103/PhysRevA.103.052408.
- [20] Jessica Lemieux, Bettina Heim, David Poulin, Krysta Svore, and Matthias Troyer. Efficient quantum walk circuits for Metropolis–Hastings algorithm. *Quantum*, 4:287, 2020. doi:10.22331/q-2020-06-29-287.
- [21] Michael Lewin, Dror Livnat, and Uri Zwick. Improved rounding techniques for the MAX 2-SAT and MAX DI-CUT problems. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 67–82. Springer, 2002. doi:10.1007/3-540-47867-1\_6.
- [22] Chu-Min Li and Felip Manyà. Theory and Applications of Satisfiability Testing–SAT 2021. 2009. doi:10.1007/978-3-030-80223-3.
- [23] Guang Hao Low and Isaac L. Chuang. Optimal Hamiltonian Simulation by Quantum Signal Processing. *Physical Review Letters*, 118:010501, Jan 2017. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.118.010501>, doi:10.1103/PhysRevLett.118.010501.
- [24] Guang Hao Low and Isaac L. Chuang. Hamiltonian Simulation by Qubitization. *Quantum*, 3:163, 2019. doi:10.22331/q-2019-07-12-163.
- [25] Jarrod R. McClean, Sergio Boixo, Vadim N. Smelyanskiy, Ryan Babbush, and Hartmut Neven. Barren plateaus in quantum neural network training landscapes. *Nature communications*, 9(1):1–6, 2018. doi:10.1038/s41467-018-07090-4.
- [26] Roberto Oliveira and Barbara M Terhal. The complexity of quantum spin systems on a two-dimensional square lattice. *arXiv:quant-ph/0504050*, 2005. URL: <https://arxiv.org/abs/quant-ph/0504050>.
- [27] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J. Love, Alán Aspuru-

- Guzik, and O'Brien Jeremy L. A variational eigenvalue solver on a photonic quantum processor. *Nature Communications*, 5(1):4213, 2014. URL: <https://doi.org/10.22331/10.1038/ncomms5213>, doi:10.1038/ncomms5213.
- [28] David Poulin, Alexei Kitaev, Damian S. Steiger, Matthew B. Hastings, and Matthias Troyer. Quantum Algorithm for Spectral Measurement with a Lower Gate Count. *Physical review letters*, 121:010501, Jul 2018. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.121.010501>, doi:10.1103/PhysRevLett.121.010501.
- [29] Jérémie Roland and Nicolas J. Cerf. Quantum Search by Local Adiabatic Evolution. *Physical Review A*, 65:042308, Mar 2002. URL: <https://link.aps.org/doi/10.1103/PhysRevA.65.042308>, doi:10.1103/PhysRevA.65.042308.
- [30] Christof Zalka. Grover's quantum searching algorithm is optimal. *Physical Review A*, 60:2746–2751, Oct 1999. URL: <https://link.aps.org/doi/10.1103/PhysRevA.60.2746>, doi:10.1103/PhysRevA.60.2746.

# Conclusion

*The nineteenth century was known as the machine age, the twentieth century will go down in history as the information age. I believe the twenty-first century will be the quantum age*  
— Davies, 1997 [40]

Au cours de cette thèse, nous avons présenté trois algorithmes et les ressources nécessaires pour obtenir les résultats escomptés avec une grande fidélité. Ces algorithmes sont tous basés sur une discrétisation de l'évolution adiabatique. Nous avons observé que l'usage heuristique des méthodes offre une réduction du coût moyen des algorithmes. En effet, une discrétisation moins dense, correspondant à une évolution *rapide* dans le cas analogique, réduit considérablement les ressources en conservant une grande probabilité de succès.

L'évolution adiabatique discrète présentée au chapitre 2 démontre bien le gain attribuable à des méthodes astucieuses comme la qubitisation. En effet, l'efficacité de la procédure combinée à l'agrandissement du gap est probablement ce qui offre le plus grand gain : jusqu'à six ordres de grandeur pour des réseaux d'une cinquantaine de sites. Le gap passe de  $\Delta$  à  $\arccos(1 - \Delta/2\pi)$  grâce à la qubitisation. De là, le coût des ressources diminue grandement, puisque l'algorithme d'évolution adiabatique ainsi que l'estimation de phase nécessite un nombre d'opérations variant selon l'inverse du gap. En annexe de l'article, nous avons présenté la tendance pour les chaînes (1D) et les réseaux échelles ( $m \times 2$ ). Dans les deux cas, la tendance s'avère inférieure aux bornes théoriques de l'évolution adiabatique. De plus, l'impact sur les réseaux échelles, un problème plus difficile que celui 1D, semble supérieur, c'est-à-dire qu'on y note une croissance des ressources plus lente. Rappelons que ces tendances sont en

fonction du gap initial, ce qui illustre l'attrait des procédures non physiques comme la qubitisation. Ces procédures sont possibles pour le modèle de circuit et non pas pour un appareil analogique.

La procédure de rembobinage, quant à elle, offre un gain moins grand (un peu plus d'un ordre de grandeur). Nous pourrions être tentés de critiquer l'usage de cette méthode puisqu'il s'agit d'une simulation longue plutôt que de plusieurs simulations courtes, ce qui devrait nécessiter l'usage de codes de correction d'erreurs plus coûteux. Or, les mesures projectives évitent l'accumulation d'erreurs. Ainsi, la procédure ne devrait pas augmenter le coût d'implémentation.

L'algorithme de marche quantique présenté dans cette thèse semble offrir un avantage sur le cas classique et sur l'opérateur de Szegedy. En effet, l'opérateur présenté est en quelque sorte la racine de l'opérateur de Szegedy, ce qui permet une réduction du coût par un facteur deux. On observe également des accélérations sous-quadratiques et super-quadratiques comparativement au cas classique.

Nous avons également utilisé une évolution adiabatique rapide, en évitant la méthode de randomisation de phases [27]. Au moment de soumettre l'article, nous ne comprenions pas exactement pourquoi nous avons une bonne convergence sans cette méthode qui imite l'effet de la mesure projective. L'article suivant, sur l'algorithme d'évolution adiabatique basé sur les réflexions, offre une explication. En effet, l'opérateur de marche n'est pas un unitaire quelconque, c'est une réflexion. Celle-ci permet de parcourir le chemin adiabatique, ou le recuit simulé, plus rapidement que la mesure projective.

L'algorithme d'évolution adiabatique basé sur les réflexions semble prometteur. En effet, le gain sur la probabilité de succès est très grand, même pour de petits problèmes (moins de 14 variables) comparativement à l'algorithme de Grover. On observe également un gain du temps de solution (TTS) malgré le coût élevé des réflexions. En combinant des procédures telle que la qubitisation, nous devrions pouvoir élargir le gap et réduire le nombre d'opérations nécessaires aux réflexions, qui sont très coûteuses. Évidemment, nous ne bénéficierons pas du fait que les mesures projectives évitent l'accumulation d'erreur au cours du calcul. Par contre, n'utilisant plus de mesures projectives, les étapes intermédiaires ne peuvent guère échouer.

En somme, le coût réel des algorithmes est très élevé. J'ai souvent entendu que l'ordinateur quantique sera majoritairement composé de modules de corrections d'erreurs entrecoupés de quelques calculs seulement. La nécessité de la correction d'erreur entraîne un surcoût majeur et est donc l'une des entraves principales à l'obtention d'un appareil démontrant la suprématie quantique. Étant donné la rapidité des processeurs classiques, les instances qui offriront un gain pratique sur les algorithmes classiques sont hors d'atteinte pour le moment. Les optimisations présentées dans cette thèse démontrent la possibilité de réduire considérablement les ressources nécessaires aux calculs quantiques, et donc la possibilité d'avoir des applications pratiques, surpassant les capacités des ordinateurs classiques, plus rapidement que les prévisions actuelles. Le défi de construire un ordinateur quantique qui n'aurait pas besoin d'autant de corrections d'erreurs persiste.

*If nature can do it, we can do it too*

— Grover, 2021 [39]

## Annexe A

# Matériel supplémentaire

## A.1 Algorithme d'estimation de phase

L'algorithme d'estimation de phase quantique (EPQ) permet d'extraire les valeurs propres d'un opérateur et de distinguer ses états propres. Prenons, par exemple, un unitaire exprimé sous la forme  $U = e^{iH}$ , pour un hamiltonien  $H$ . La valeur propre de l'unitaire, associée au vecteur propre  $|j\rangle$ , est donnée par une expression de la forme  $e^{iE_j}$ , où  $E_j$  la  $j$ ème valeur propre de  $H$ . Rappelons que  $H$  et  $U$  ont les mêmes vecteurs propres. L'EPQ appliquée à un état quelconque  $|\psi\rangle$  permet d'obtenir la superposition

$$\text{EPQ}|\psi\rangle = \sum_j \langle j|\psi\rangle |E_j/2\pi\rangle |j\rangle.$$

Dû à la périodicité de l'exponentielle complexe, les valeurs propres doivent satisfaire  $0 \leq E_j < 2\pi$  pour tout  $j$  afin de pouvoir être distinguées clairement les unes des autres. Ainsi, une mesure de l'énergie provoque l'effondrement de la fonction d'onde dans l'état propre associé.

L'EPQ se base sur deux principes : le retour de phase et la transformée de Fourier quantique.

**Retour de phase** Comme  $|j\rangle$  est état propre de  $U$ , appliquer  $U$  à  $|j\rangle$  ne fait qu'ajouter une phase. Ainsi, si l'on contrôle l'opération par un qubit dans l'état  $|+\rangle$ , la phase n'apparaîtra que lorsque le qubit de contrôle est dans l'état  $|1\rangle$ . C'est ce qu'on appelle le retour de phase, puisque la phase semble être appliquée au qubit de contrôle. Par exemple, supposons que  $U|j\rangle = -|j\rangle$ , alors :

$$(C - U)|+\rangle |j\rangle = |-\rangle |j\rangle, \tag{A.1}$$

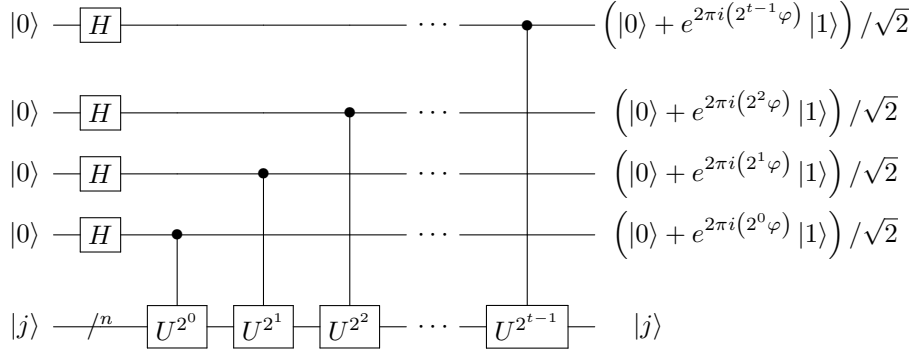


FIGURE A.1 – Première étape de l'estimation de phase

la notation  $C-U$  désigne l'opération contrôle- $U$ . De manière générale, nous pouvons écrire la valeur propre comme étant  $e^{2\pi i\varphi}$ . Ainsi, nous avons :

$$(C-U)|+\rangle|j\rangle = (|0\rangle + e^{2\pi i\varphi}|1\rangle)|j\rangle. \quad (\text{A.2})$$

D'où on peut conclure :

$$(C-U)^{2^k}|+\rangle|j\rangle = (|0\rangle + e^{2\pi i2^k\varphi}|1\rangle)|j\rangle. \quad (\text{A.3})$$

La première étape de l'algorithme est illustrée à la Fig. A.1. Le retour de phase se fait en appliquant une tour de Hadamard au premier registre, le registre ancillaire,  $|0\rangle^{\otimes t} \xrightarrow{H^{\otimes t}} |+\rangle^{\otimes t} = \sum_{k=0}^{2^t-1} |k\rangle$  puis en appliquant une série de  $C-U$  à la puissance  $2^k$  où  $k$  va de  $t$  à  $0$ , du premier au dernier qubit ancillaire respectivement. L'état à la fin de cette première étape est donc [13] :

$$\frac{1}{2^{t/2}}(|0\rangle + e^{2\pi i2^{t-1}\varphi}|1\rangle)(|0\rangle + e^{2\pi i2^{t-2}\varphi}|1\rangle)\dots(|0\rangle + e^{2\pi i2^0\varphi}|1\rangle) = \frac{1}{2^{t/2}} \sum_{k'=0}^{2^t-1} e^{2\pi i\varphi k'} |k'\rangle \quad (\text{A.4})$$

Si  $\varphi$  peut s'écrire exactement avec  $t$  bits, alors on exprime  $\varphi = \sum_{k=1}^t \varphi_k 2^{-k}$ , ou de manière abrégée,  $\varphi = 0.\varphi_1\varphi_2\dots\varphi_t$ . Notons que si  $\varphi$  ne peut s'écrire exactement avec  $t$  bits, alors nous avons une approximation de  $\varphi$  à une précision  $2^{-t}$ . L'approximation doit être suffisamment précise pour distinguer deux états rapprochés, c'est-à-dire qu'elle doit être plus précise que la différence des valeurs propres. On peut récrire l'équation précédente avec cette nouvelle notation :

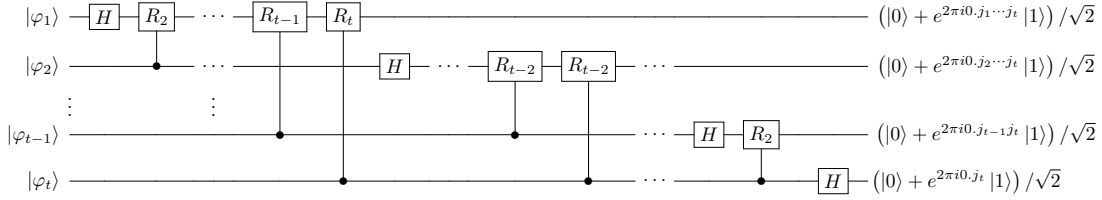


FIGURE A.2 – Transformée de Fourier quantique

$$\frac{1}{2^{t/2}} (|0\rangle + e^{2\pi i 0 \cdot \varphi_t} |1\rangle) (|0\rangle + e^{2\pi i 0 \cdot \varphi_{t-1} \varphi_t} |1\rangle) \dots (|0\rangle + e^{2\pi i 0 \cdot \varphi_1 \dots \varphi_{t-1} \varphi_t} |1\rangle) \quad (\text{A.5})$$

L'entièreté de l'information qui nous intéresse est contenue dans les phases de l'état, c'est pourquoi nous allons faire appel à la transformée de Fourier inverse.

**Transformée de Fourier quantique** La transformée de Fourier quantique (TFQ) est identique à la transformée de Fourier discrète. Cette dernière permet de passer d'un vecteur de nombres complexes  $(x_0, \dots, x_{N-1})$  à un autre  $(y_0, \dots, y_{N-1})$  où

$$y_{k'} \equiv \frac{1}{\sqrt{N}} \sum_{j'=0}^{N-1} x_{j'} e^{2\pi i j' k' / N}. \quad (\text{A.6})$$

De manière équivalente, la TFQ sur un état arbitraire est définie comme

$$\sum_{j'=0}^{N-1} x_{j'} |j'\rangle \rightarrow \sum_{k'=0}^{N-1} y_{k'} |k'\rangle$$

où  $y_{k'}$  sont données par la transformée de Fourier discrète de  $x_{j'}$  [13]. Sur des vecteurs de base  $|0\rangle, \dots, |N-1\rangle$ , l'effet est directement

$$|j'\rangle \rightarrow \frac{1}{\sqrt{N}} \sum_{k'=0}^{N-1} e^{2\pi i j' k' / N} |k'\rangle.$$

On remarque que la TFQ permet de transférer l'information contenue dans la base de l'état vers ses amplitudes. L'algorithme est coûteux. Pour  $t$  qubits,  $O(t^2)$  contrôles-opérations sont nécessaires. Toutefois, les opérations sur la phase sont en général plus efficaces que sur l'état du qubit ce qui rend, entre autres, cette sous-routine avantageuse ; elle permet le changement de base entre la base de calcul et la "base de Fourier".



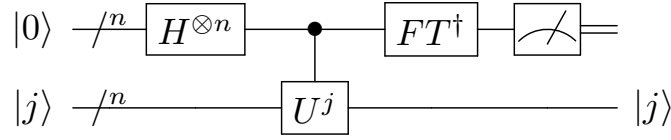


FIGURE A.3 – Estimation de phase quantique

Le circuit quantique de la TFQ est illustré à la Fig. A.2. Les rotations

$$R_k = \begin{bmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^k} \end{bmatrix} \quad (\text{A.7})$$

L'EPQ, illustrée à la Fig. A.3, permet ainsi d'extraire l'information relative à la phase grâce à une série d'opérations dans l'espace de Fourier. En effet, la tour de Hadamard coïncide exactement à la TFQ sur l'état zéro. Puis, le retour de phase permet l'obtention de l'information relative aux valeurs propres. Cette information, encodée dans les amplitudes relatives des qubits ancillaires, est transférée dans l'état des qubits grâce au changement de base effectué par la TFQ inverse.

## A.2 Le modèle de Hubbard

Hubbard a proposé un modèle approximatif décrivant l'interaction d'électrons de bandes étroites. Le modèle décrit, avec une mathématique plus simple que celle décrivant exactement l'interaction de Coulomb, les matériaux où les interactions entre électrons sont importantes [31]. On réduit souvent la complexité du modèle en ne considérant qu'une orbitale par site plutôt que la structure électronique complète [41].

Il existe quatre états possibles par site :

$$|\text{Vide}\rangle = |0\rangle, \quad |\uparrow\rangle, \quad |\downarrow\rangle, \quad \text{et} \quad |\uparrow\downarrow\rangle.$$

L'espace de Hilbert croît donc comme  $4^n$  où  $n$  est le nombre de sites. Le hamiltonien,  $H$ , est défini comme :

$$H = \sum_{\sigma\langle i,j\rangle} (T_{ij}c_{i\sigma}^\dagger c_{j\sigma} + T_{ji}^*c_{j\sigma}^\dagger c_{i\sigma}) + U \sum_i n_{i\uparrow}n_{i\downarrow} \quad (\text{A.8})$$

L'opérateur  $c_{i\sigma}^\dagger$  représente l'opérateur de création au site  $i$  d'un fermion de spin  $\sigma = \{\uparrow, \downarrow\}$  par opposition à l'opérateur d'annihilation  $c_{i\sigma}$ . La notation  $\langle i, j \rangle$  souligne que le terme est autorisé uniquement entre deux sites adjacents (premier voisin).

$$c_\sigma |\sigma\rangle = |0\rangle \qquad c_\sigma^\dagger |\sigma\rangle = 0 \qquad (\text{A.9})$$

$$c_\sigma |0\rangle = 0 \qquad c_\sigma^\dagger |0\rangle = |\sigma\rangle \qquad (\text{A.10})$$

$$(\text{A.11})$$

Ils obéissent aux relations suivantes :

$$\{c_i, c_j^\dagger\} = c_i c_j^\dagger + c_j^\dagger c_i = \delta_{ij} \qquad (\text{A.12})$$

$$\{c_i, c_j\} = 0 \qquad (\text{A.13})$$

$$\{c_i^\dagger, c_j^\dagger\} = 0 \qquad (\text{A.14})$$

La matrice  $T$  appelée matrice de saut, décrit les amplitudes des transitions possibles entre les sites. Par exemple,  $T_{ij} = 0$  implique qu'un déplacement du site  $i$  au site  $j$  d'une particule est impossible. Le premier terme de A.8 exprime donc l'énergie cinétique (déplacement des fermions).

L'opérateur  $n_{i\sigma} = c_{i\sigma}^\dagger c_{i\sigma}$  représente l'occupation du site  $i$  par un fermion de spin  $\sigma$  et donc  $n_{i\uparrow} n_{i\downarrow}$  représente la double occupation du site  $i$ .

$$n_\sigma |0\rangle = 0 \qquad n_\sigma |\sigma\rangle = |\sigma\rangle \qquad n_\sigma |\uparrow\downarrow\rangle = |\uparrow\downarrow\rangle \qquad (\text{A.15})$$

$$n_\uparrow n_\downarrow |0\rangle = 0 \qquad n_\uparrow n_\downarrow |\sigma\rangle = 0 \qquad n_\uparrow n_\downarrow |\uparrow\downarrow\rangle = |\uparrow\downarrow\rangle \qquad (\text{A.16})$$

Ainsi, en comptant le nombre de doubles occupations, on obtient l'énergie potentielle (l'interaction entre les fermions) dont l'échelle est fixée par  $U$ . On remarque que l'interaction entre particules se limite à une courte distance — sur un même site pour être exact.

**Intérêt et plage d'application** À demi-remplissage, le modèle décrit le comportement d'un isolant de Mott [41]. Un isolant de Mott est un matériau typiquement conducteur qui possède une phase isolante causée par l'interaction électron-électron.

En effet, sous certains paramètres, la forte interaction causée par  $U$  induit un gap dans la bande originale d'énergie, ce qui donne naissance à cette phase isolante [42]. Dans le cas décrit par le modèle de Hubbard à demi-remplissage et à  $U = 0$ , l'état fondamental est un métal. La phase isolante apparaît avec  $U > 0$ .

Le modèle de Hubbard est d'un grand intérêt en physique de la matière condensée et est l'objet d'études approfondies depuis déjà plusieurs décennies. En effet, dans la limite thermodynamique, le modèle en deux dimension peut décrire des phénomènes d'intérêts, dont l'antiferromagnétisme [43] et la supraconductivité [44].

Le modèle de Hubbard est particulièrement difficile à résoudre lorsque les énergies cinétique et coulombienne sont en compétition. L'effet de ces interactions est maximal à demi-remplissage ou à proximité du demi-remplissage. Même numériquement, le problème demeure difficile à résoudre.

### A.3 Marche aléatoire classique

Les méthodes de Monte-Carlo, de manière générale, sont une classe d'algorithmes où on procède par échantillonnage pour calculer numériquement la moyenne d'une observable dans un espace trop grand pour permettre un calcul exact, ou systématique. Il existe une multitude d'applications de ces méthodes, la plus connue étant certainement l'intégration numérique. Les méthodes de Monte-Carlo permettent donc d'obtenir un échantillon de variables aléatoires indépendantes  $\{X_1, \dots, X_n\}$  et identiquement distribuées selon une densité  $f$ . Par la loi des grands nombres, la moyenne des résultats obtenus à partir d'un grand nombre d'essais devrait être proche de la moyenne théorique [45].

Il existe différentes classes de méthodes d'échantillonnage à partir d'une distribution de probabilité, dont les méthodes de Monte-Carlo par chaîne de Markov (MCMC). Une chaîne de Markov est un processus stochastique obéissant à la propriété de Markov : la distribution conditionnelle de probabilité des états futurs est donnée par l'état actuel uniquement ; la distribution conditionnelle est indépendante des états passés. Une chaîne de Markov *irréductible*<sup>1</sup> possède une distribution stationnaire unique représentée par le vecteur colonne  $\pi$  :

$$P\pi = \pi, \tag{A.17}$$

Les chaînes de Markov utilisées dans les méthodes de MCMC sont choisies de manière à ce que leur distribution stationnaire coïncide avec la distribution de probabilité à échantillonner. Les méthodes les plus simples utilisent des marches aléatoires sur ces

---

1. Une chaîne de Markov est dite *irréductible* si tout état  $y$  est accessible par tout état  $x$

chaînes. La marche aléatoire est donc un processus stochastique de modélisation. On appelle *pas*, l'itération consistant à prendre un échantillon au hasard sur la chaîne de Markov. La *direction*<sup>2</sup> est choisie selon la distribution conditionnelle de probabilité des états futurs. L'échantillonnage forme un chemin, une séquence de pas, dans un espace mathématique décrivant le problème.

Une caractéristique importante des chaînes de Markov est le temps de mélange (de l'anglais « mixing time »), soit le temps requis (nombre de pas) pour atteindre la distribution stationnaire. Celui-ci est régi par l'inverse du gap spectral de  $P$ , soit la différence entre ses deux plus grandes valeurs propres. La distribution stationnaire d'une chaîne de Markov correspond à l'état propre associé à la valeur propre 1 de la matrice de transition. Intuitivement, on comprend qu'en appliquant à répétition la matrice de transition  $P$  à une distribution initiale quelconque, le poids des états propres correspondant aux valeurs propres plus petites que 1 diminuera progressivement. Le temps de mélange correspond donc au nombre de pas nécessaire pour obtenir des réalisations indépendantes de la distribution de probabilité initiale, c'est-à-dire pour obtenir la distribution stationnaire. Plus le gap spectral est petit, plus long sera le processus. Pour une revue détaillée des processus stochastiques, tels que les chaînes de Markov, consulter [36, 46].

On peut visualiser la matrice de transition d'une chaîne de Markov grâce à un graphe où chaque noeud représente un état de l'espace et où les probabilités de transition d'un état à un autre sont représentées par des flèches pondérées. Le concept de direction est représenté naturellement par la flèche, nous indiquant la possibilité de passer d'un état A à un état B, sans pour autant garantir la possibilité de passer de l'état B à l'état A avec la même probabilité. La pondération dicte la probabilité de transition de l'état courant vers un autre, ou la probabilité qu'il y demeure. Ainsi, la somme des probabilités de quitter un état ou d'y rester doit être égale à 1. Pour un exemple, voir Fig. A.4.

Afin de mieux comprendre le détail de ces méthodes ainsi que la discrétisation quantique que nous allons en faire, penchons-nous sur une méthode de MCMC particulière, soit l'algorithme de Métropolis-Hastings [38].

Il est facile d'imaginer un espace des états qui croît de manière exponentielle avec la taille du problème. C'est le cas notamment pour les chaînes de spin. Dans un tel cas, il est inconcevable de générer et de stocker l'entièreté de la matrice de transition de la chaîne de Markov avant de procéder à l'échantillonnage. C'est entre autres pourquoi il est d'usage d'avoir une méthode permettant le calcul en temps réel de ces probabilités de transitions. On peut également se demander pourquoi il serait efficace de calculer des probabilités de transition en temps réel, mais qu'il en serait autrement

---

2. Le terme *direction* est utilisé ici pour décrire l'éventail de possibilités d'états ou d'éléments accessibles à partir de l'état courant.



- (b) Calculer le rapport d'acceptance  $\alpha = f(y)/f(x)$
- (c) Générer un nombre aléatoire sur une distribution uniforme  $u \in [0, 1]$
- (d) Si  $u \leq \alpha$ , accepter le déplacement, c.-à-d.  $x_{t+1} = y$
- (e) Si  $u > \alpha$ , refuser le déplacement, c.-à-d.  $x_{t+1} = x_t$

Comme  $f(x)$  est proportionnelle à la densité  $P(x)$ , nous avons que  $f(x)/f(y) = P(x)/P(y)$ . L'algorithme de Métropolis-Hastings permet ainsi de calculer en temps réel les probabilités de transitions sans utiliser la distribution globale  $P$ .

## A.4 Opérateur de marche de Szegedy

Szegedy a introduit en 2004 une méthode générique permettant la quantification des algorithmes basés sur les marches aléatoires classiques [37]. Sous certaines conditions, notamment lorsque la chaîne de Markov est ergodique (apériodique et irréductible) et que la matrice de transition est symétrique, la marche quantique offre une accélération quadratique sur son homologue classique. Il s'agit alors d'une généralisation de l'algorithme de Grover. Les résultats de Szegedy démontrent que, tout comme le cas classique, le *temps de frappe*<sup>3</sup> quantique dépend du gap spectral de la matrice de transition.

L'opérateur de marche de Szegedy fait une quantification des marches sur des graphes *bipartis*<sup>4</sup>. Définissons un graphe biparti fini dont les sous-ensembles distincts sont  $X$  et  $Y$ . Soient  $P = (p_{x,y})$  et  $Q = (q_{y,x})$  des matrices décrivant des probabilités de transitions de  $X$  à  $Y$  et de  $Y$  à  $X$  respectivement<sup>5</sup>. La marche sur le graphe  $X \cup Y$ , définie par les matrices de transitions  $P$  et  $Q$ , forme une marche classique sur un graphe biparti. Ainsi,  $p_{x,y}, q_{y,x} \geq 0$  et

$$\sum_{y \in Y} p_{x,y} = 1 \quad \forall x \in X, \tag{A.20}$$

$$\sum_{x \in X} q_{y,x} = 1 \quad \forall y \in Y. \tag{A.21}$$

---

3. Le *temps de frappe* (de l'anglais « hitting time ») est le nombre de pas moyen pour atteindre un état  $y$  à partir d'un état  $x$ .

4. Les graphes sont dits *bipartis* si l'ensemble de ses noeuds peuvent être divisé en deux sous-ensembles disjoints.

5. Les choix de notation sont faits de manière semblable à l'article original de Szegedy, mais en utilisant la notation de Dirac.

Notons que toute marche aléatoire classique (notamment les chaînes de Markov) peut devenir "bipartite" en dupliquant les états, c'est-à-dire en posant  $X = Y$  et  $P = Q$ . La quantification se fait en définissant deux opérateurs dans un espace de Hilbert définie par les bases

$$\{|x\rangle |y\rangle |x \in X, y \in Y\} \quad (\text{A.22})$$

Les états

$$|\phi_x\rangle = \sum_{y \in Y} \sqrt{p_{x,y}} |x\rangle |y\rangle, \quad \text{et} \quad (\text{A.23})$$

$$|\psi_y\rangle = \sum_{x \in X} \sqrt{q_{y,x}} |x\rangle |y\rangle \quad (\text{A.24})$$

permettent de définir l'opérateur de marche de Szegedy,  $W = R_2 R_1$ , où

$$R_1 = 2 \sum_{x \in X} |\phi_x\rangle \langle \phi_x| - \mathbb{1}, \quad (\text{A.25})$$

$$R_2 = 2 \sum_{y \in Y} |\psi_y\rangle \langle \psi_y| - \mathbb{1}. \quad (\text{A.26})$$

Dans le cas classique, les états occupés sont définis et l'élément aléatoire vient des probabilités de transitions entre les états. On obtient des réalisations décorréliées de la distribution initiale en prenant un nombre de pas suffisamment grand, un phénomène expliqué par le temps de mélange. Dans le cas quantique, c'est différent. On initialise l'ordinateur dans un état où les amplitudes correspondent à la racine des probabilités données par la distribution initiale. Les opérations quantiques, telles que l'opérateur de Szegedy, permettent une évolution unitaire, donc non aléatoire. C'est la mesure qui induit l'élément de hasard nécessaire à l'échantillonnage en provoquant l'effondrement de la fonction d'onde.

Dans les marches aléatoires, le temps de frappe est lié de près au temps requis pour visiter tous les états possibles. Tout comme le temps de mélange pour le cas classique, le temps de frappe varie en fonction de l'inverse du gap spectral de la matrice de transition. Szegedy a démontré que le temps de frappe de sa quantification correspond à la racine du gap spectral classique. L'opérateur de marche quantique permet donc de traverser le domaine, l'espace des états, en un temps qui est quadratiquement plus court que sa contrepartie classique. Toutes les directions sont explorées d'un coup, en superposition, et selon la racine de la probabilité. Ceci est essentiellement dû au fait que l'opérateur quantique est défini en fonction d'amplitudes plutôt que de probabilités. Ainsi, on peut trouver un ensemble d'éléments cibles, tout comme dans l'algorithme de Grover, quadratiquement plus rapidement que dans le cas classique. Notons que l'algorithme de Grover peut être interprété comme une marche aléatoire sur un graphe complet.

# Bibliographie

- [1] David Poulin. Quantum information science. Dans *Aspen Physics*, (2018). [cf. p. 1]
- [2] Calcul Québec. Histoire du calcul informatique de pointe, (2017). [cf. p. 1]
- [3] Gordon E. Moore. Cramming more components onto integrated circuits, reprinted from electronics, volume 38, number 8, april 19, 1965, pp.114 ff. *IEEE Solid-State Circuits Society Newsletter* **11(3)**, 33–35 (2006). [cf. p. 1]
- [4] TOP500 Supercomputer Database. Supercomputer power (flops), 1993 ro 2020. *Our World in Data* (2020). [cf. p. 2]
- [5] B H Bransden et Joachain. *Quantum Mechanics, second edition*. Pearson Prentice Hall, (2000). [cf. p. 3]
- [6] Richard P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics* **21**, 467–488 (1982). [cf. p. 3]
- [7] Joseph W Britton, Brian C Sawyer, Adam C Keith, C-C Joseph Wang, James K Freericks, Hermann Uys, Michael J Biercuk et John J Bollinger. Engineered two-dimensional ising interactions in a trapped-ion quantum simulator with hundreds of spins. *Nature* **484(7395)**, 489–492 (2012). [cf. p. 3]
- [8] Lov K Grover. A fast quantum mechanical algorithm for database search. Dans *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, 212–219, (1996). [cf. p. 4, 15]
- [9] Peter W Shor. Algorithms for quantum computation : discrete logarithms and factoring. Dans *Proceedings 35th annual symposium on foundations of computer science*, 124–134. Ieee, (1994). [cf. p. 4]
- [10] Nikolaj Moll, Panagiotis Barkoutsos, Lev S Bishop, Jerry M Chow, Andrew Cross, Daniel J Egger, Stefan Filipp, Andreas Fuhrer, Jay M Gambetta, Marc Ganzhorn et others. Quantum optimization using variational algorithms on near-term quantum devices. *Quantum Science and Technology* **3(3)**, 030503 (2018). [cf. p. 4]
- [11] John Preskill. Quantum computing in the nisq era and beyond. *Quantum* **2**, 79 (2018). [cf. p. 4]
- [12] Seth Lloyd. Quantum algorithm for solving linear systems of equations. Dans *APS March Meeting Abstracts*, tome 2010, D4–002, (2010). [cf. p. 4]



- [13] Michael A Nielsen et Isaac L Chuang. Quantum computation and quantum information. *Phys. Today* **54**(2), 60 (2001). [cf. p. [6](#), [69](#), [70](#)]
- [14] Sergey Bravyi et Jeongwan Haah. Magic-state distillation with low overhead. *Phys. Rev. A* **86**(5), 052329 (2012). [cf. p. [9](#)]
- [15] Neil J Ross et Peter Selinger. Optimal ancilla-free clifford+ t approximation of z-rotations. *arXiv preprint arXiv :1403.2975* (2014). [cf. p. [10](#)]
- [16] Dorit Aharonov, Wim Van Dam, Julia Kempe, Zeph Landau, Seth Lloyd et Oded Regev. Adiabatic quantum computation is equivalent to standard quantum computation. *SIAM review* **50**(4), 755–787 (2008). [cf. p. [11](#), [14](#)]
- [17] Krishanu Sankar, Artur Scherer, Satoshi Kako, Sam Reifenstein, Navid Ghardemarzy, Willem B Krayenhoff, Yoshitaka Inui, Edwin Ng, Tatsuhiko Onodera, Pooya Ronagh et others. Benchmark study of quantum algorithms for combinatorial optimization : Unitary versus dissipative. *arXiv preprint arXiv :2105.03528* (2021). [cf. p. [11](#)]
- [18] Troels F Rønnow, Zhihui Wang, Joshua Job, Sergio Boixo, Sergei V Isakov, David Wecker, John M Martinis, Daniel A Lidar et Matthias Troyer. Defining and detecting quantum speedup. *science* **345**(6195), 420–424 (2014). [cf. p. [11](#), [12](#)]
- [19] Ryan Babbush, Jarrod R McClean, Michael Newman, Craig Gidney, Sergio Boixo et Hartmut Neven. Focus beyond quadratic speedups for error-corrected quantum advantage. *PRX Quantum* **2**(1), 010103 (2021). [cf. p. [12](#)]
- [20] Tameem Albash et Daniel A Lidar. Adiabatic quantum computation. *Reviews of Modern Physics* **90**(1), 015002 (2018). [cf. p. [13](#), [14](#)]
- [21] Jérémie Roland et Nicolas J. Cerf. Quantum search by local adiabatic evolution. *Phys. Rev. A* **65**, 042308 (2002). [cf. p. [14](#), [19](#)]
- [22] Edward Farhi, Jeffrey Goldstone, Sam Gutmann et Michael Sipser. Quantum computation by adiabatic evolution. *arXiv preprint quant-ph/0001106* (2000). [cf. p. [14](#)]
- [23] Christof Zalka. Grover’s quantum searching algorithm is optimal. *Phys. Rev. A* **60**(4), 2746 (1999). [cf. p. [15](#), [18](#)]
- [24] Gilles Brassard et Peter Hoyer. An exact quantum polynomial-time algorithm for simon’s problem. Dans *Proceedings of the Fifth Israeli Symposium on Theory of Computing and Systems*, 12–23. IEEE, (1997). [cf. p. [15](#)]
- [25] Wim Van Dam, Michele Mosca et Umesh Vazirani. How powerful is adiabatic quantum computation? Dans *Proceedings 42nd IEEE symposium on foundations of computer science*, 279–287. IEEE, (2001). [cf. p. [19](#)]
- [26] Sabine Jansen, Mary-Beth Ruskai et Ruedi Seiler. Bounds for the adiabatic approximation with applications to quantum computation. *J. Math. Phys.* **48**(10), 102111 (2007). [cf. p. [19](#)]

- [27] Sergio Boixo, Emanuel Knill, Rolando D Somma et others. Eigenpath traversal by phase randomization. *Quantum Inf. Comput.* **9**(9&10), 833–855 (2009). [cf. p. [21](#), [22](#), [37](#), [66](#)]
- [28] Dorit Aharonov et Amnon Ta-Shma. Adiabatic quantum state generation. *SIAM Journal on Computing* **37**(1), 47–82 (2007). [cf. p. [22](#)]
- [29] Carlton M Caves et Christopher A Fuchs. Quantum information : How much information in a state vector? *arXiv preprint quant-ph/9601025* (1996). [cf. p. [25](#)]
- [30] Matthias Troyer et Uwe-Jens Wiese. Computational complexity and fundamental limitations to fermionic quantum monte carlo simulations. *Physical review letters* **94**(17), 170201 (2005). [cf. p. [25](#)]
- [31] John Hubbard. Electron correlations in narrow energy bands. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* **276**(1365), 238–257 (1963). [cf. p. [26](#), [71](#)]
- [32] Ulrich Schollwöck. The density-matrix renormalization group in the age of matrix product states. *Annals of physics* **326**(1), 96–192 (2011). [cf. p. [26](#)]
- [33] Román Orús. A practical introduction to tensor networks : Matrix product states and projected entangled pair states. *Annals of Physics* **349**, 117–158 (2014). [cf. p. [26](#)]
- [34] Chris Marriott et John Watrous. Quantum arthur–merlin games. *computational complexity* **14**(2), 122–152 (2005). [cf. p. [26](#)]
- [35] Guang Hao Low et Isaac L Chuang. Hamiltonian simulation by qubitization. *Quantum* **3**, 163 (2019). [cf. p. [26](#)]
- [36] Samuel Karlin. *A first course in stochastic processes*. Academic press, (2014). [cf. p. [36](#), [74](#)]
- [37] Mario Szegedy. Quantum speed-up of markov chain based algorithms. Dans *45th Annual IEEE symposium on foundations of computer science*, 32–41. IEEE, (2004). [cf. p. [36](#), [76](#)]
- [38] W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika* **57**(1), 97–109 (1970). [cf. p. [37](#), [74](#)]
- [39] Lov K. Grover. Is quantum searching a universal property of nature? Electrical Engineering and Sense, Collect and Move Center Seminar, Columbia University, Data Science Institute, (2021). [cf. p. [53](#), [67](#)]
- [40] Gerard J Milburn. *Schrodinger’s machines : the quantum technology reshaping everyday life*. Henry Holt and Company, (1997). [cf. p. [65](#)]
- [41] Dave Wecker, Matthew B Hastings, Nathan Wiebe, Bryan K Clark, Chetan Nayak et Matthias Troyer. Solving strongly correlated electron models on a quantum computer. *Phys. Rev. A* **92**(6), 062318 (2015). [cf. p. [71](#), [72](#)]
- [42] Masaki Uchida. *Spectroscopic study on charge-spin-orbital coupled phenomena in Mott-Transition oxides*. Springer Science & Business Media, (2013). [cf. p. [73](#)]

- [43] Kenn Kubo et Mamoru Uchinami. The antiferromagnetic ground state of a half-filled hubbard model. *Progress of Theoretical Physics* **54**(5), 1289–1298 (1975). [cf. p. 73]
- [44] Th. Maier, M. Jarrell, Th. Pruschke et J. Keller. *d*-wave superconductivity in the hubbard model. *Phys. Rev. Lett.* **85**, 1524–1527 (2000). [cf. p. 73]
- [45] FM Dekking C Kraaikamp et HP Lopuhaä LE Meester. A modern introduction to probability and statistics, (2005). [cf. p. 73]
- [46] Don Van Ravenzwaaij, Pete Cassey et Scott D Brown. A simple introduction to markov chain monte-carlo sampling. *Psychonomic bulletin & review* **25**(1), 143–154 (2018). [cf. p. 74]